Implementation of Fast Fourier Transform Algorithms with the TMS32020

APPLICATION REPORT: SPRA122

Authors:

Panos Papamichalis
Digital Signal Processing – Semiconductor Group

John So
Atlanta Regional Technology

Digital Signal Processing Solutions
1989

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain application using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

TRADEMARKS

CONTACT INFORMATION

US TMS320 HOTLINE	(281) 274-2320
US TMS320 FAX	(281) 274-2324
US TMS320 BBS	(281) 274-2323
US TMS320 email	dsph@ti.com

Implementation of Fast Fourier Transform Algorithms with the TMS32020

Abstract

Fast Fourier Transforms (FFT), containing a class of computationally efficient algorithms implementing the Discrete Fourier Transforms (DFT), are widely used in DSP applications. In this report on FFT, the development of the FFT from the continuous Fourier Transform and DFT is first presented. Issues regarding the implementation of the FFT with the TMS32020 processor are then discussed, such as scaling, special FFT structures, and system memory and I/O considerations. The report also includes the TMS32020 code for 256-point and 1024-point FFT algorithms.

Product Support on the World Wide Web

Our World Wide Web site at www.ti.com contains the most up to date product information, revisions, and additions. Users registering with TI&ME can build custom information pages and receive new product updates automatically via email.

INTRODUCTION

The Fourier transform converts information from the time domain into the frequency domain. It is an important analytical tool in such diverse fields as acoustics, optics, seismology, telecommunications, speech, signal processing, and image processing. In discrete-time systems, the Discrete Fourier Transform (DFT) is the counterpart of the continuous-time Fourier transform. Since the DFT is computation-intensive, it had relatively few applications, even with modern computers. The Fast Fourier Transform (FFT) is the generic name for a class of computationally efficient algorithms that implement the DFT and are widely used in the field of Digital Signal Processing (DSP).

Recent advances in VLSI hardware, such as the Texas Instruments TMS320 family of digital signal processors, have further enhanced the popularity of the FFT. This application report describes the implementation of FFT algorithms using the TMS32020 processor, which has features particularly suited to digital signal processing. This report begins with a discussion of the development of the DFT algorithm, leading to the derivation of the FFT algorithm. Special attention is given to various FFT implementation aspects, such as scaling. Although this report refers to radix-2 and radix-4 FFT only, the implementation techniques described are applicable to all FFT algorithms in general.[1-3] Specific examples of FFT implementations on the TMS32010 processor are contained in the book by Burrus and Parks.[4] To expedite TMS32020 FFT code development, two macro libraries are included in the appendices for both the direct and indirect memory addressing modes. TMS32020 source code examples are also given for a 256-point (both radix-2 and radix-4) and a 1024-point complex FFT, along with some system memory considerations for implementing large FFTs. The FFT source code can be found in Appendices A through G.

DEVELOPMENT OF THE DFT ALGORITHM

The Discrete Fourier Transform (DFT) is the discrete-time version of the continuous-time Fourier transform. The continuous-time Fourier transform or frequency spectrum of an analog signal x(t) is

$$X(w) = \int_{-\infty}^{\infty} x(t)e^{-jwt}dt \tag{1}$$

where, in general, both x(t) and X(w) are complex functions of the continuous-time variable t and the frequency variable w, respectively. The continuous-time signal x(t) is converted to a discrete-time signal x(nT) by sampling it every T seconds. When there is no ambiguity, the sampling period T notation is dropped and the discrete signal is represented by x(n). The Fourier transform of the discrete signal is given by

$$X(w) = \sum_{n=-\infty}^{\infty} x(n)e^{-jwn} \tag{2}$$

where w represents normalized frequency and takes on values between 0 and 2π. X(w) is periodic with period 2π and, as a result, it is sufficient to consider its values only between 0 and 2π.[2] The periodicity of X(w) is a direct result of the sampled nature of x(n). In general, sampling in the time domain is associated with periodicity in the frequency domain and, conversely, sampling in the frequency domain is associated with periodicity in the time domain. This property is a basic result in Fourier theory, and forms the foundation of the DFT.

Assume that a signal x(n) consists of N samples. Since no restriction is imposed on what happens outside the interval of N points, it is convenient to assume that the signal is periodically repeated. Under this assumption, and because of the above correspondence of sampling and periodicity, the Fourier transform becomes discrete with the distance between successive samples equal to the fundamental frequency of the signal in the time domain. This distance is $2\pi/N$ in normalized frequency units. The result is the DFT, given by

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{nk} \qquad k = 0,1,\ldots,N-1 \tag{3}$$

where $W_N = e^{-j2\pi/N}$, and W_N is known as the phase or twiddle factor. Equation (3) is generally referred to as an N-point DFT. Because the number of complex multiplications and additions required is approximately N^2 for large N, the total number of arithmetic operations required for a given N increases rapidly with the value of N. In fact, the excessively large amount of computations required to compute the DFT directly when N is large has directly prompted alternative methods for computing the DFT efficiently. Most of these methods make use of the inherent symmetry and periodicity of the above twiddle factor, as shown in Figure 1 for the case where $N=8$.

Figure 1 shows that the following symmetry and periodicity relationships are true:

$$\text{Symmetry Property:} \quad W_N^k = -W_N^{k+(N/2)} \tag{4}$$

$$\text{Periodicity Property:} \quad W_N^k = W_N^{N+k} \tag{5}$$

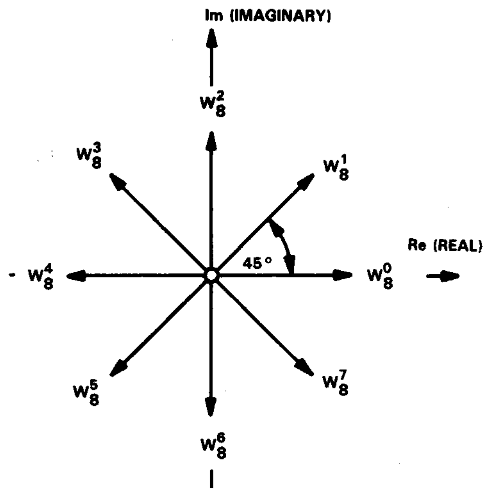

Figure 1. Symmetry and Periodicity of the Twiddle Factor for N=8

In the next section, these relationships are utilized in the derivation of the radix-2 FFT algorithm.

DERIVATION OF THE FFT ALGORITHM

A more efficient method of computing the DFT that significantly reduces the number of required arithmetic operations is the so-called decimation-in-time (DIT) FFT algorithm.[2] With the FFT, N is a factorable number that allows the overall N-point DFT to be decomposed into successively smaller and smaller transforms. The size of the smallest transform thus derived is known as the radix of the FFT algorithm. Thus, for a radix-2 FFT algorithm, the smallest transform or "butterfly" (basic computational unit) used is the 2-point DFT. Generally, for an N-point FFT, there are N resultant frequency samples corresponding to N time samples of the input signal x(n). For a radix-2 FFT, N is a power of 2.

The number of arithmetic operations can be reduced initially by decomposing the N-point DFT into two N/2-point DFTs. This means that the input time sequence x(n) is decomposed into two N/2-point subsequences (hence the name, decimation-in-time), which consist of its even-numbered and odd-numbered samples with time indices expressed mathematically as 2n and 2n+1, respectively. Substituting these time indices into the original DFT equation gives

$$X(k) = \sum_{n=0}^{N/2-1} x(2n)\, W_N^{2nk} + \sum_{n=0}^{N/2-1} x(2n+1)\, W_N^{(2n+1)k} \tag{6}$$

$$= \sum_{n=0}^{N/2-1} x(2n)\, W_N^{2nk} + W_N^k \sum_{n=0}^{N/2-1} x(2n+1)\, W_N^{2nk}$$

Since

$$W_N^2 = [e^{-j(2\pi/N)}]^2 = [e^{-j\pi/(N/2)}]^2 = W_{N/2}$$

equation (6) can be written as

$$X(k) = \sum_{n=0}^{N/2-1} x(2n)\, W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} x(2n+1)\, W_{N/2}^{nk} \tag{7}$$

$$= Y(k) + W_N^k\, Z(k) \qquad k = 0,1,\ldots,N-1$$

where Y(k) is the first summation term and Z(k) is the second summation term.

Y(k) and Z(k) are further seen to be the N/2-point DFTs of the even-numbered and odd-numbered time samples, respectively. In this case, the number of complex multiplications and additions is approximately $N + 2(N/2)^2$ because, according to (7), the N-point DFT is split in two N/2-point DFTs, which are then combined by N complex multiplications and additions. Thus, by splitting the original N-point DFT into two N/2-point DFTs, the total number of arithmetic operations has been reduced. This reduction is illustrated in Figure 2.

Implicit in the above derivation is the periodicity of X(k), Y(k), and Z(k). X(k) is periodic in k with a period N, while Y(k) and Z(k) are both periodic in k with a period N/2. Consequently, despite the fact that the index k ranges over N values from 0 to $N-1$ for X(k), both Y(k) and Z(k) must be computed for k between 0 and $(N/2)-1$ only. The periodicity of Y(k) and Z(k) is also assumed in Figure 2.

Although (7) can be used to evaluate X(k) for $0 \le k \le N-1$, further reduction in the amount of computation is possible when the symmetry property (4) and periodicity (5) of the twiddle factor are utilized to compute X(k) separately over the following ranges:

1st Half of Frequency Spectrum: $0 \le k \le (N/2)-1$

2nd Half of Frequency Spectrum: $(N/2) \le k \le (N-1)$

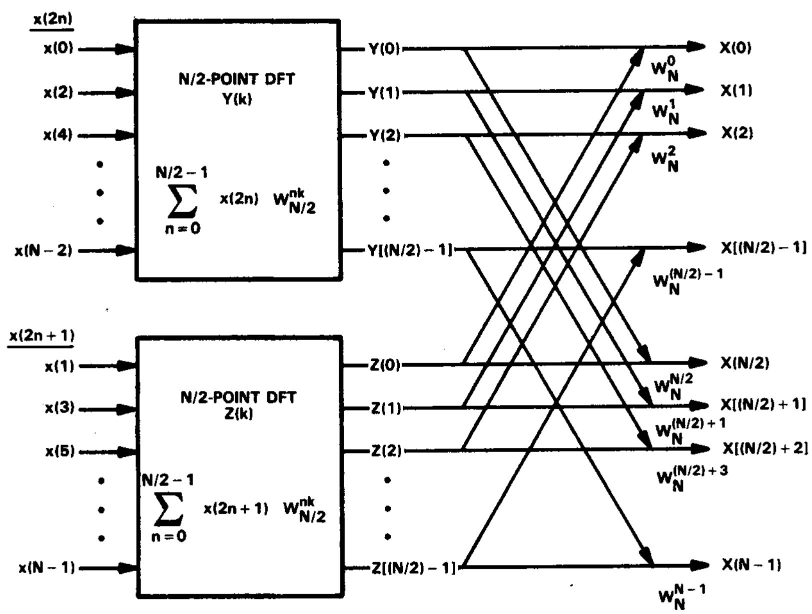

Figure 2. First DIT Decomposition of an N-Point DFT

Equation (7), for $N/2 \le k \le N-1$, can be rewritten as

$$X(k+N/2) = \sum_{n=0}^{N/2-1} x(2n)\, W_{N/2}^{n(k+N/2)} + W_N^{k+N/2} \sum_{n=0}^{N/2-1} x(2n+1)\, W_{N/2}^{n(k+N/2)} \quad (8)$$

$$\text{where } 0 \le k \le (N/2)-1$$

Since

$$W_{N/2}^{n(k+N/2)} = W_{N/2}^{n(N/2)}\, W_{N/2}^{nk} = e^{-j2\pi n}\, W_{N/2}^{nk} = W_{N/2}^{nk}$$

and

$$W_N^{k+N/2} = W_N^k\, e^{-j\pi} = -W_N^k$$

equation (8) can be rewritten as

$$X(k+N/2) = \sum_{n=0}^{N/2-1} x(2n)\, W_{N/2}^{nk} - W_N^k \sum_{n=0}^{N/2-1} x(2n+1)\, W_{N/2}^{nk} \quad (9)$$

$$= Y(k) - W_N^k Z(k) \qquad k = 0,1,\ldots,(N/2)-1$$

Therefore, (7) can be used to compute the first half of the frequency spectrum X(k) for the index range $0 \le k \le (N/2)-1$, while equation (9) can be used to compute the second half of the frequency spectrum $X(k+N/2)$.

Figure 3 depicts the situation when the symmetry property of the twiddle factor is used to compute X(k). The above decimation process and symmetry exploitation can reduce the DFT computation tremendously. By further decimating the odd-numbered and even-numbered time samples in a similar fashion, four N/4-point DFTs can be obtained, resulting in a further reduction in the DFT computation. Consequently, to arrive at the final radix-2 DIT FFT algorithm, this decimation process is repetitively carried out until eventually the N-point DFT can be evaluated as a collection of 2-point DFTs or butterflies.

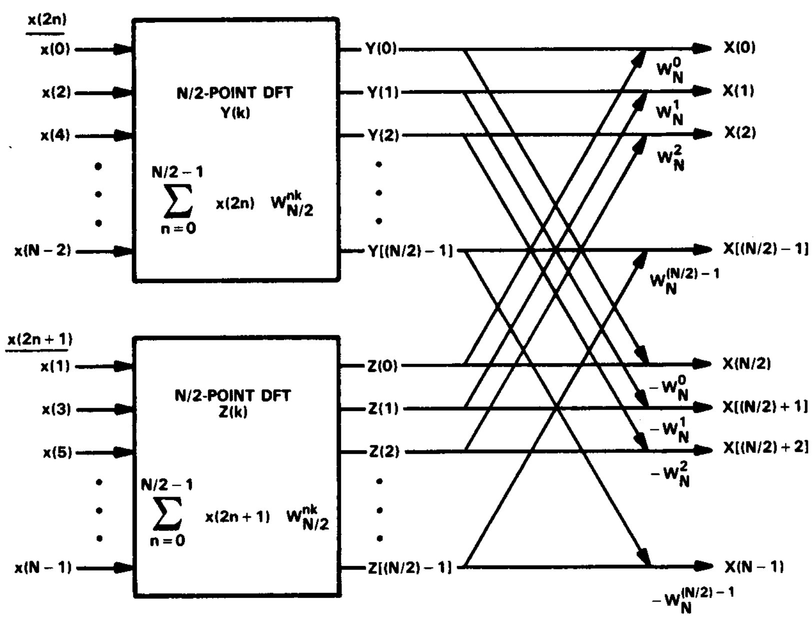

Figure 3. Decomposition of a DFT Using the Symmetry Property

RADIX-2 DECIMATION-IN-TIME (DIT) FFT BUTTERFLY

In the radix-2 DIT FFT algorithm, the time decimation process passes through a total of M stages where $N = 2^M$ with N/2 2-point FFTs or butterflies per stage, giving a total of $(N/2)\log_2 N$ butterflies per N-point FFT.

For the case of an 8-point DFT implemented using the radix-2 DIT FFT algorithm discussed in the previous pages, the input samples are processed through three stages. Four butterflies are required per stage, giving a total of twelve butterflies in the radix-2 implementation. Each butterfly is a 2-point DFT of the form depicted in Figure 4. P and Q are the inputs to the radix-2 DIT FFT butterfly. In general, the inputs to each butterfly are complex as is also the twiddle factor.

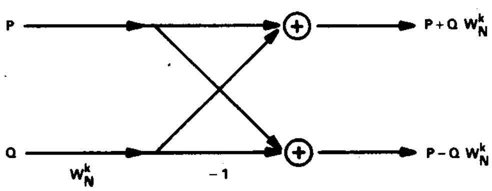

Figure 4. A Radix-2 DIT FFT Butterfly Flowgraph

As shown in Figure 4, the outputs P' and Q' of the radix-2 butterfly are given by

$$P' = P + Q\,W_N^k$$
$$Q' = P - Q\,W_N^k \qquad (10)$$

While Figure 4 actually uses signal flowgraph nomenclature, another commonly used symbol for a radix-2 butterfly is shown in Figure 5.

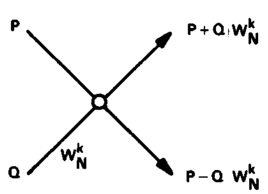

Figure 5. A Simplified Radix-2, DIT FFT Butterfly Symbol

For an explanation of the various notational conventions in use, the reader is referred to reference [3]. Both the flowgraph nomenclature and the butterfly symbol are used interchangeably in this report.

Implementation of the FFT Butterfly with Scaling

In the computation of the FFT, scaling of the intermediate results becomes necessary to prevent overflows. The TMS32020 processor has features optimized for digital signal processing and a number of on-chip shifters for scaling. In particular, the input scaling shifter, the 32-bit double-precision ALU and accumulator, and its output shifters are used extensively for scaling.

To see why scaling is necessary, observe that from the general equation of an N-point DFT (3), application of Parseval's theorem gives

$$\sum_{n=0}^{N-1} x^2(n) = \frac{1}{N}\sum_{k=0}^{N-1} |X(k)|^2 \qquad (11)$$

or

$$N\left[\frac{1}{N}\sum_{n=0}^{N-1} x^2(n)\right] = \left[\frac{1}{N}\sum_{k=0}^{N-1} |X(k)|^2\right] \qquad (12)$$

i.e., the mean-squared value of X(k) is N times that of input x(n). Consequently, in computing the DFT of the input sequence x(n), overflows may occur when fixed-point arithmetic is employed without appropriate scaling. To see how overflows can actually occur in FFT computations, consider the general radix-2 butterfly in the mth stage of an N-point DIT FFT as shown in Figure 6.

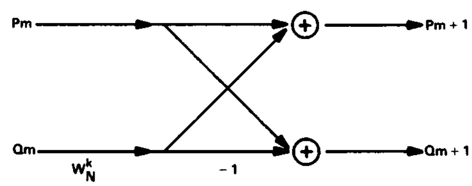

Figure 6. Signal Flowgraph of a Butterfly at the mth Stage

From Figure 6, the final form of the FFT can be written as

$$Pm+1 = Pm + W_N^k\,Qm$$
$$Qm+1 = Pm - W_N^k\,Qm \qquad (13)$$

where Pm and Qm are the inputs, and Pm + 1 and Qm + 1 are the outputs of the mth stage of the N-point FFT, respectively. In general, Pm, Qm, Pm + 1, and Qm + 1 are complex as is the twiddle factor. The twiddle factor can be expressed as

$$W_N^k = e^{-j(2\pi/N)k} = \cos(X) - j\sin(X) \qquad (14)$$

where $X = (2\pi/N)k$ and $j = \sqrt{-1}$.

The inputs Pm and Qm can be expressed in terms of their real and imaginary parts by

$$\begin{aligned} Pm &= PR + j\,PI \\ Qm &= QR + j\,QI \end{aligned} \qquad (15)$$

By substituting the values from (14) and (15), equation (13) becomes

$$\begin{aligned} Pm+1 &= PR + j\,PI + (QR\cos(X) + QI\sin(X) \\ &\quad + j\,(QI\cos(X) - QR\sin(X)) \\ &= (PR + QR\cos(X) + QI\sin(X)) \\ &\quad + j\,(PI + QI\cos(X) - QR\sin(X)) \end{aligned} \qquad (16)$$

$$\begin{aligned} Qm+1 &= PR + j\,PI - (QR\cos(X) + QI\sin(X)) \\ &\quad - j\,(QI\cos(X) - QR\sin(X)) \\ &= (PR - QR\cos(X) - QI\sin(X)) \\ &\quad + j\,(PI - QI\cos(X) + QR\sin(X)) \end{aligned}$$

Although the inputs of each butterfly stage have real and imaginary parts with magnitudes less than one, the real and imaginary parts of the outputs from (15) can have a maximum magnitude of

$$1 + 1\sin(45) + 1\cos(45) = 2.414213562$$

To avoid the possibility of overflow, each stage of the FFT is scaled down by a factor of 2. In this way, if an FFT consists of M stages, the output is scaled down by $2^M = N$, where N is the length of the FFT. Even with scaling, overflow is possible because of the maximum magnitude value for complex input data. This possibility is avoided by scaling down the input signal by a factor of 1.207106781, and then scaling up the output of the last FFT stage by the same factor. This additional scaling is not implemented in the code of the appendices, because the input signal is assumed real (i.e., the imaginary part is zero), and the above maximum value cannot be attained. The maximum value for a real input is 2.

Using (15), the TMS32020 butterfly code is given in Figure 7. It is assumed that all input and output data values are in Q15 format; i.e., they are expressed in two's-complement fractional arithmetic with the binary point immediately to the right of the sign bit (15 bits after the binary point). This code incorporates one stage of scaling (i.e., scaling by two) for the implementation of the general radix-2 DIT FFT butterfly with the 16-bit sine and cosine values of the twiddle factor also stored in Q15 format. Note that in performing fractional multiplications, the product of two 16-bit Q15 fractions is a 32-bit double-precision fraction in Q30 format with two sign bits. This result is illustrated in Figure 8, where S stands for sign bit.

The code for a general radix-2 DIT FFT is given in Figure 7. In the comment section, ACC, P-REGISTER, and T-REGISTER represent the on-chip 32-bit accumulator, 32-bit product register, and 16-bit temporary register of the TMS32020 processor, respectively. For more information about the TMS32020 processor and its architecture, see the TMS32020 User's Guide.[5]

The first block in the butterfly code of Figure 7 (starting with the label INIT) is for general system initialization. The second block of code (starting with the label BTRFLY) takes advantage of the double sign bits to provide a "free" divide-by-2 scaling in calculating the term (1/2)(QR COS(X) + QI SIN(X)), which is the scaled real part of the product of the twiddle factor and Qm. In addition, since the current contents of memory location QR are no longer required for subsequent calculations, QR is also used as a temporary storage for this term.

The third block of code calculates the term (1/2) (QI COS(X) − QR SIN(X)), which is the scaled imaginary part of the product of the twiddle factor and Qm. By completing this calculation, QI is also freed as a temporary storage for this term.

The fourth block of code calculates the real parts of Pm + 1 and Qm + 1 and provides the divide-by-2-per-stage scaling function to avoid signal overflows. To perform this function, the input binary scaling shifter of the TMS32020 is used.

```
*****************************************************************
* TMS32020 CODE FOR A GENERAL RADIX-2 DIT FFT BUTTERFLY         *
*****************************************************************
*
* EQUATES FOR THE REAL AND IMAGINARY PARTS OF Xm(P) AND Xm(Q).
* THE LOCATIONS PR, PI, QR, AND QI ARE USED BOTH FOR THE INPUT
* AND THE OUTPUT DATA.
*
PR      EQU     0       Re(Pm) STORED IN LOCATION 0 IN DATA MEMORY
PI      EQU     1       Im(Pm) STORED IN LOCATION 1 IN DATA MEMORY
QR      EQU     2       Re(Qm) STORED IN LOCATION 2 IN DATA MEMORY
QI      EQU     3       Im(Qm) STORED IN LOCATION 3 IN DATA MEMORY
*
* EQUATES FOR THE REAL AND IMAGINARY PARTS OF THE TWIDDLE FACTOR.
*
COSX    EQU     4       COS(X) STORED IN LOCATION 4 IN DATA MEMORY
SINX    EQU     5       SIN(X) STORED IN LOCATION 5 IN DATA MEMORY
*
* INITIALIZE SYSTEM.
*
        AORG    >20
INIT    SPM     0       NO SHIFT AT OUTPUTS OF P-REGISTER
        SSXM            SELECT SIGN-EXTENSION MODE
        ROVM            RESET OVERFLOW MODE
        LDPK    4       CHOOSE DATA PAGE 4
*
* CALCULATE (QR COS(X) + QI SIN(X)); STORE RESULT IN QR.
*
BTRFLY LT       QR      LOAD T-REGISTER WITH QR
        MPY     COSX    P-REGISTER = (1/2) QR COSX
        LTP     QI      ACC= (1/2) QR COSX ; LOAD T-REGISTER WITH QI
        MPY     SINX    P-REGISTER = (1/2) QI SINX
        APAC            ACC= (1/2)(QR COSX+QI SINX)
        MPY     COSX    P-REGISTER = (1/2) QI COSX
        LT      QR      LOAD T-REGISTER WITH QR
        SACH    QR      QR = (1/2)(QR COSX+QI SINX)
*
* CALCULATE (QI COS(X) - QR SIN(X)); STORE RESULT IN QI.
*
        PAC             ACC= (1/2) QI COSX
        MPY     SINX    P-REGISTER = (1/2) QR SINX
        SPAC            ACC= (1/2)(QI COSX - QR SINX)
        SACH    QI      QI = (1/2)(QI COSX - QR SINX)
*
* CALCULATE Re(Pm+1) AND Re(Qm+1); STORE RESULTS IN PR AND QR.
*
        LAC     PR,14   ACC= (1/4)PR
        ADD     QR,15   ACC= (1/4)(PR + QR COSX + QI SINX)
        SACH    PR,1    PR = (1/2)(PR + QR COSX + QI SINX)
        SUBH    QR      ACC= (1/4)(PR - QR COSX - QI SINX)
        SACH    QR,1    QR = (1/2)(PR - QR COSX - QI SINX)
*
```

Figure 7. TMS32020 code for a General Radix-2 DIT FFT Butterfly

```
* CALCULATE Im[Pm+1] AND Im[Qm+1]; STORE RESULTS IN PI AND QI.
*
        LAC     PI,14     ACC= (1/4)PI
        ADD     QI,15     ACC= (1/4)(PI + QI COSX - QR SINX)
        SACH    PI,1      PI = (1/2)(PI + QI COSX - QR SINX)
        SUBH    QI        ACC= (1/4)(PI - QI COSX + QR SINX)
        SACH    QI,1      QI = (1/2)(PI - QI COSX + QR SINX)
*
```

Figure 7. TMS32020 Code for a General Radix-2 DIT FFT Butterfly (concluded)

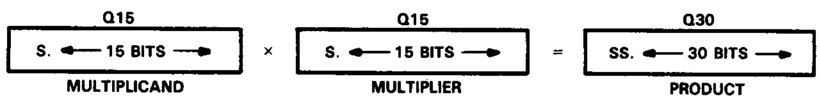

Figure 8. Multiplication of Two Q15 Numbers

Initially, the contents of PR are scaled down by a factor of 4 (equivalent to a 14-bit left-shift). Note that the shift or scaling function is being performed while the contents of PR are being loaded into the 32-bit accumulator. Since the TMS32020 has a 32-bit double-length accumulator, no accuracy is lost in this binary scaling process. To generate the final result of Re(Pm + 1), the contents of QR must be added to the contents of the accumulator with a 1-bit right-shift (equivalent to a 15-bit left-shift). This means adding (1/4)(QR COS(X) + QI SIN(X)) to (1/4)PR, which is the current value held in the accumulator. The upper-half of the accumulator is then stored in PR with a 1-bit left-shift to yield the term (1/2)(PR + QR COS(X) + QI SIN(X)), which is precisely Re(Pm + 1) scaled down by 2. This shift or scaling function is being performed while the contents of the upper half of the accumulator are loaded into PR. At this point, the accumulator still has a value equal to (1/4)(PR + QR COS(X) + QI SIN(X)). Hence, to obtain the final result of Re(Qm + 1), the unscaled contents of QR must be subtracted from the accumulator. The upper-half of the accumulator is again stored in QR with a 1-bit left-shift to yield the term (1/2)(PR + QR COS(X) + QI SIN(X)), which is precisely Re(Qm + 1) scaled down by 2.

In a similar fashion, the fifth block of code calculates the imaginary parts of Pm + 1 and Qm + 1. Note that all the scaling functions performed so far have come "free" with the architecture of the TMS32020.[5]

In summary, the data values are scaled down by right-shifting the 16-bit words as they are loaded into the 32-bit accumulator. In this way, full precision is still maintained in all calculations. The right-shifts are implemented by a corresponding number of left-shifts into the upper half of the accumulator. On the other hand, if the accumulator had been single precision or 16 bits wide, all scaling operations would have resulted in a loss of accuracy.

In-Place FFT Computations

In the butterfly implementation, the set of input registers in data memory (PR, PI, QR, and QI) for the two complex inputs Pm and Qm are used for holding the two complex outputs Pm + 1 and Qm + 1, respectively. When the same set of input registers is used as output registers for holding the FFT results, the FFT computation is said to be performed in-place. Therefore, FFTs implemented on the TMS32020 using the general butterfly routine are performed in-place.

As a general rule, an in-place FFT computation means that a total of 2N memory locations are required for an N-point FFT since the inputs to the FFT can be complex. On the other hand, a total of up to 4N memory locations is required for not-in-place computations.

Another attractive feature of the butterfly routine is that temporary or scratch-pad registers are not needed for intermediate results or calculations. Where coefficient quantization and other finite wordlength effects are not critical, 13-bit sine and cosine values can be used instead of 16-bit values addressed by the MPY instruction. In this way, the registers COSX and SINX for the twiddle factors can be dispensed with altogether. For this purpose, the MPYK instruction, which has a 13-bit signed constant embedded in its opcode, can be employed instead of the MPY instruction in the butterfly code. In Appendix A, two FFT macros (NORM1 and NORM2) illustrate the use of the MPY and MPYK instructions, respectively. Appendices A and B contain macro libraries that perform the same tasks, but in Appendix A they use direct addressing while in Appendix B they use indirect addressing.

Bit-Reversal/Data Scrambling

As shown in Figure 9, the input time samples x(n) are not in order, i.e., they are scrambled. Such data scrambling or bit reversal is a direct result of the radix-2 FFT derivation. On closer inspection, it is seen that the index of each input sample is actually bit-reversed, as shown in Table 1.

Therefore, the input data sequence must be prescrambled prior to executing the FFT in order to produce in-order outputs. To perform bit reversal on the 8-point FFT, shown in Figure 9, the pairs of input samples, [x(1) and x(4)] and [x(3) and x(6)], must be swapped. On the other hand, Figure 10 has in-order input samples by rearranging the ordering of all the butterflies. However, the outputs are now bit-reversed.

Table 1. Bit-Reversal Algorithm for an 8-Point Radix-2 DIT FFT

INDEX	BIT PATTERN	BIT-REVERSED PATTERN	BIT-REVERSED INDEX
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

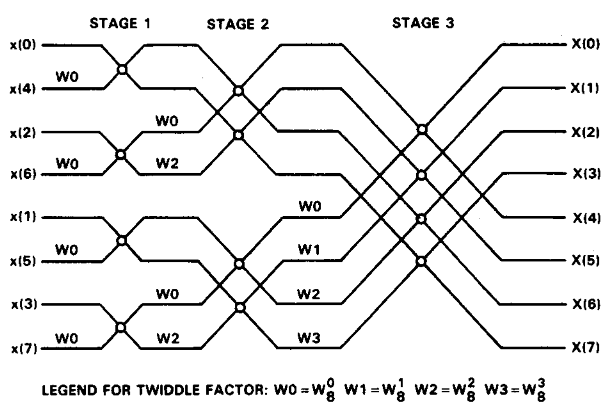

Figure 9. An In-Place DIT FFT with In-Order Outputs and Bit-reversed Inputs

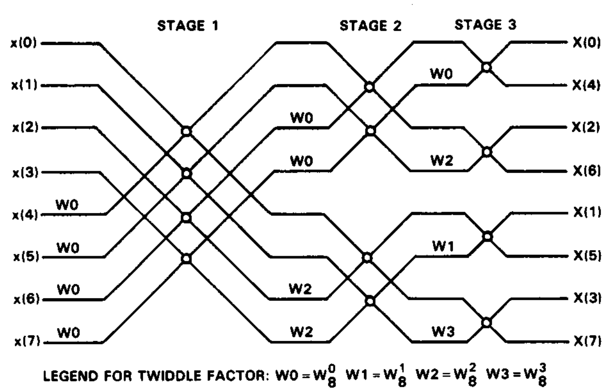

Figure 10. An In-Place DIT FFT with In-Order Inputs but Bit-Reversed Outputs

In general, bit reversal or data scrambling must be performed either at the input stage on the time samples (Figure 9) or at the output stage on the frequency samples (Figure 10). Bit reversal can be performed in-place. Such a process generally requires the use of one temporary data memory location.

Because of its double-precision accumulator and its versatile instruction set, the TMS32020 processor can perform in-place bit reversal or data scrambling without the use of a temporary data memory location. For example, the TMS32020 code for swapping input data locations x(1) and x(4) is given in Figure 11.

Although bit-reversal can be regarded as a separate task performed either at the input or output stage of an FFT implementation, some FFT algorithms exist with bit reversal as an integral part.[5] Such algorithms are said to be in-place and in-order, and they tend to have higher execution speeds than that of the FFT and bit-reversal algorithms executed separately.

A Numerical Example: 8-Point DIT FFT

To illustrate the concept of the FFT, a numerical example of an 8-point, decimation-in-time FFT is presented. The input signal is a square pulse with four samples equal to 0.5 and four samples equal to zero, as shown in Figure 12(a). The broken line in Figure 12 represents the envelope of the plotted signal. Figure 12(b) plots the magnitude of the computed FFT, where the number next to each sample indicates its magnitude. The choice of the amplitude for this example is arbitrary, but it is restricted to be less than 1 since it assumed that the numbers handled by the processor are in Q15 format.

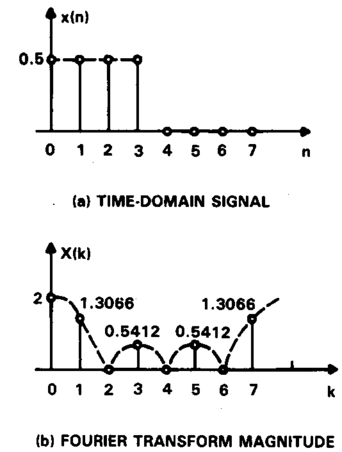

Figure 12. Time-Domain Signal and the Magnitude of Its FFT

The FFT of this time signal is computed by an 8-point DIT FFT as shown in Figure 13. On the left side, the samples x(n) of the time signal are arranged in their normal order. On the right side, the computed samples X(k) of the FFT are in bit-reversed order. Since the computation produces complex numbers, all the numerical values are presented as (R, I), where R is the real part and I is the imaginary part of the complex number. Figure 13 shows also the numerical values computed in the intermediate stages.

```
*********************************************************************
* TMS32020 CODE FOR THE BIT REVERSAL OF x(1) AND x(4)              *
*********************************************************************
*
BITREV ZALH   RX1     LOAD REAL PART OF x(1) IN UPPER ACCUMULATOR
       ADDS   RX4     LOAD REAL PART OF x(4) IN LOWER ACCUMULATOR
       SACL   RX1     STORE REAL PART OF x(4) IN REAL PART OF x(1)
       SACH   RX4     STORE REAL PART OF x(1) IN REAL PART OF x(4)
       ZALH   IX1     LOAD IMAG PART OF x(1) IN UPPER ACCUMULATOR
       ADDS   IX4     LOAD IMAG PART OF x(4) IN LOWER ACCUMULATOR
       SACL   IX1     STORE IMAG PART OF x(4) IN IMAG PART OF x(1)
       SACH   IX4     STORE IMAG PART OF x(1) IN IMAG PART OF x(4)
```

Figure 11. TMS32020 Code for the Bit Reversal of x(1) and x(4)

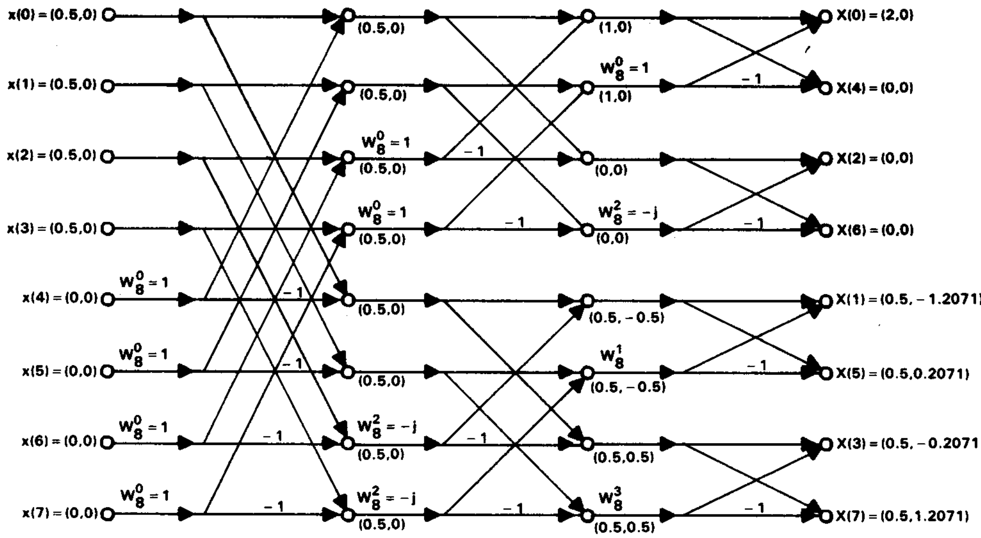

Figure 13. Numerical Example of an 8-Point DIT FFT without Scaling

Table 2 shows the values of the twiddle factor W_8^i for $i = 0,1,\ldots,7$. Of these factors, only the first four are used in the FFT. The other four are related to them through the symmetry property (see equation (4)).

Table 2. Numerical Values of W_8^i, where i = 0, 1, ..., 7

TWIDDLE FACTOR	VALUE
W_8^0	1
$W_8^1 = e^{-\pi/4}$	0.7071 − j 0.7071
$W_8^2 = e^{-\pi/2}$	−j
$W_8^3 = e^{-3\pi/4}$	−0.7071 − j 0.7071
$W_8^4 = e^{-\pi} = -W_8^0$	−1
$W_8^5 = e^{-5\pi/4} = -W_8^1$	−0.7071 + j 0.7071
$W_8^6 = e^{-3\pi/2} = -W_8^2$	j
$W_8^7 = e^{-7\pi/4} = -W_8^3$	0.7071 + j 0.7071

Figure 13 has demonstrated the need for scaling. Without scaling, the intermediate results can attain values greater than or equal to 1. This would cause overflows in an implementation that uses Q15 numbers. Therefore, scaling is applied as mentioned earlier. Figure 14 shows exactly the same example, but now every stage is scaled by 1/2. No overflows occur with this implementation. The final output is the same as in Figure 13 but scaled by 1/8.

Special Butterflies

Although any N-point FFT (where N is a power of 2) can be directly implemented with the general butterfly only, special butterflies are normally used in order to increase the FFT execution speed.

Special butterflies can be coded by taking advantage of certain sine and cosine values of the twiddle factor. For instance, when the angle X takes on values such as 0, 90, 180, and 270 degrees, butterflies require much less code. Other special butterflies can also be coded for angles such as 45, 135, 225, and 315 degrees. Examples of these special butterflies can be found in nine macros located in Appendix A.

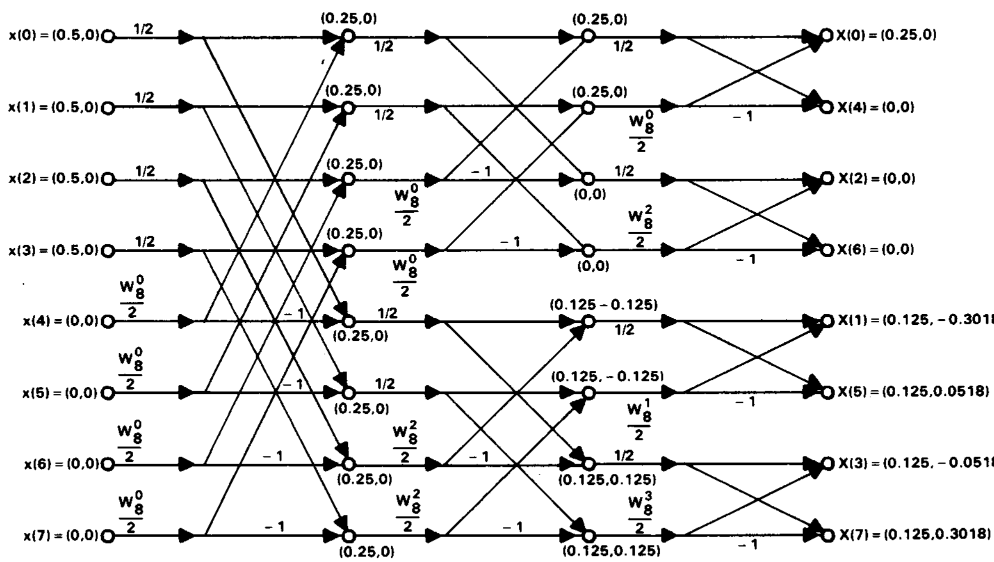

Figure 14. Numerical Example of an 8-Point DIT FFT with Scaling

An interesting point to be noted is that the first two stages of an N-point radix-2 FFT can be performed simultaneously with a special radix-4 butterfly to enhance execution speed. This special radix-4 butterfly is depicted in Figure 15 with the corresponding code (MACRO 9) listed in Appendix A.

The special radix-4 butterfly actually consists of four separate radix-2 butterflies. The radix-4 butterfly is further seen to be a 4-point DFT.

Together with the general butterfly, these special butterflies greatly improve the execution speed of an FFT algorithm. An example of the use of such butterflies for an 8-point DIT FFT is given in Figure 16. Since the FFT implementation is, in general, highly modular, the code in Figure 16 has been structured into a number of macro calls, including a macro for bit reversal.

During assembly time, the TMS32020 Macro Assembler fully expands these macros into in-line code.[5] The first two stages of the 8-point DIT FFT are implemented by the special radix-4 DIT FFT macro COMBO. The last stage consists of the special radix-2 DIT FFT macros ZERO, PIBY4, PIBY2, and PI3BY4. These macros can be found in Appendix A. The difference from the general radix-2 DIT butterfly is that the angle X of the twiddle factor takes on the values 0, 45, 90, and 135, respectively.

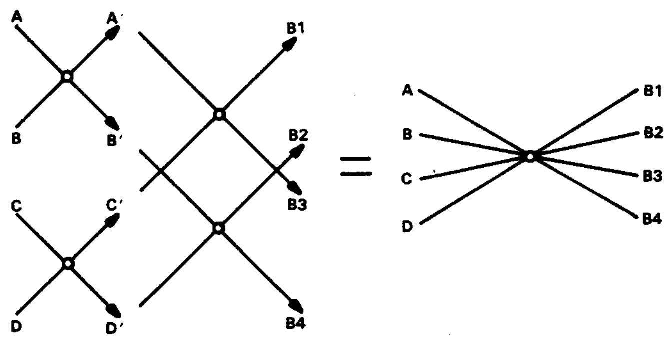

Figure 15. The Equivalence of Four Radix-2 Butterflies to one Radix-4 Butterfly

```
**********************************************************************
*              AN 8-POINT DIT FFT                                    *
**********************************************************************
*
X0R    EQU    00
X0I    EQU    01
X1R    EQU    02
X1I    EQU    03
X2R    EQU    04
X2I    EQU    05
X3R    EQU    06
X3I    EQU    07
X4R    EQU    08
X4I    EQU    09
X5R    EQU    10
X5I    EQU    11
X6R    EQU    12
X6I    EQU    13
X7R    EQU    14
X7I    EQU    15
W      EQU    16
       AORG   >20
WTABLE DATA   >5A82    VALUE FOR SIN(45) OR COS(45)
```

Figure 16. TMS32020 Code for an 8-Point DIT FFT Implementation

```
*
* INITIALIZE SYSTEM
*
INIT   SPM    0        NO SHIFT AT OUTPUTS OF PR
       SSXM            SELECT SIGN-EXTENSION MODE
       ROVM            RESET OVERFLOW MODE
       LDPK   4        CHOOSE DATA PAGE 4
       LALK   WTABLE   GET TWIDDLE FACTOR ADDRESS
       TBLR   W        STORE SIN(45) OR COS(45) IN W
*
* MACRO FOR INPUT BIT REVERSAL
*
BITREV $MACRO PR,PI,QR,QI
       ZALH   :PR:
       ADDS   :QR:
       SACL   :PR:
       SACH   :QR:
       ZALH   :
       ADDS   :QI:
       SACL   :
       SACH   :QI:
       $END
*
* FFT CODE WITH BIT-REVERSED INPUT SAMPLES
*
FFT8PT BITREV 2,3,8,9
       BITREV 6,7,12,13
*
* FIRST & SECOND STAGES COMBINED WITH DIVIDE-BY-4 INTERSTAGE SCALING
*
       COMBO  X0R,X0I,X1R,X1I,X2R,X2I,X3R,X3I
       COMBO  X4R,X4I,X5R,X5I,X6R,X6I,X7R,X7I
*
* THIRD STAGE WITH DIVIDE-BY-2 INTERSTAGE SCALING
*
       ZERO   X0R,X0I,X4R,X4I
       PIBY4  X1R,X1I,X5R,X5I,W
       PIBY2  X2R,X2I,X6R,X6I
       PI3BY4 X3R,X3I,X7R,X7I,W
```

Figure 16. TMS32020 Code for an 8-Point DIT FFT Implementation (concluded)

RADIX-4 DECIMATION-IN-FREQUENCY (DIF) FFT

The implementation described thus far is that of a radix-2 FFT using Decimation In Time (DIT). The decimation-in-time FFT is calculated by breaking the input sequence x(n) into smaller and smaller sequences and computing their FFTs. In an alternate approach, the output sequence X(k), which represents the Fourier transform of x(n), can be broken down into smaller subsequences that are computed from x(n). This method is called Decimation In Frequency (DIF). Computationally, there is no real difference between the two approaches. DIF is introduced here for two reasons: (1) to give the reader a broader understanding of the different methods used for the computation of the FFT, and (2) to allow a comparison of this implementation with the FORTRAN programs provided in the book by Burrus and Parks.[4] The programs from that book were the basis for the development of the radix-4 FFT code on the TMS32020.

In a radix-4 FFT, each butterfly has four inputs and four outputs instead of two as in the case of radix-2 FFT. As shown in the following equations, this is advantageous because the twiddle factor W has special values when the exponent corresponds to multiples of $\pi/2$. The end result is that the computational load of the FFT is reduced, and the radix-4 FFT is computed faster than the radix-2 FFT.

To introduce the radix-4 DIF FFT, equation (3) is broken into four summations. These four summations correspond to the four components in radix-4. The choice of having N/4 consecutive samples of x(n) in each sum is dictated by the choice of Decimation In Frequency (DIF).

$$
\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n)\, W_N^{nk} = \sum_{n=0}^{(N/4)-1} x(n)\, W_N^{nk} \\
&+ \sum_{n=N/4}^{(N/2)-1} x(n)\, W_N^{nk} + \sum_{n=N/2}^{(3N/4)-1} x(n)\, W_N^{nk} \\
&+ \sum_{n=3N/4}^{N-1} x(n)\, W_N^{nk} = \sum_{n=0}^{(N/4)-1} x(n)\, W_N^{nk} \\
&+ W_N^{Nk/4} \sum_{n=0}^{(N/4)-1} x(n+N/4)\, W_N^{nk} \\
&+ W_N^{Nk/2} \sum_{n=0}^{(N/4)-1} x(n+N/2)\, W_N^{nk} \\
&+ W_N^{3Nk/4} \sum_{n=0}^{(N/4)-1} x(n+3N/4)\, W_N^{nk}
\end{aligned} \tag{17}
$$

$$k = 0,1,\ldots,N-1$$

From the definition of the twiddle factor, it can be shown that

$$W_N^{Nk/4} = (-j)^k,\ W_N^{Nk/2} = (-1)^k, \text{ and } W_N^{3Nk/4} = (j)^k$$

where j is the square root of -1. With this substitution, (17) can be rewritten as

$$
\begin{aligned}
X(k) = \sum_{n=0}^{(N/4)-1} [x(n)+(-j)^k\, x(n+N/4) \\
+(-1)^k\, x(n+N/2)+(j)^k\, x(n+3N/4)]\, W_N^{nk}
\end{aligned} \tag{18}
$$

Equation (18) is not yet an FFT of length N/4, because the twiddle factor depends on N and not on N/4. To make it an N/4-point FFT, the sequence X(k) is broken into four sequences (decimation in frequency) for the cases where $k = 4r,\ 4r+1,\ 4r+2$, and $4r+3$.

Introducing this segmentation, and remembering that

$$W_N^{4nr} = W_{N/4}^{nr}$$

the following four equations (19) are derived from (18)

$$
\begin{aligned}
X(4r) &= \sum_{n=0}^{(N/4)-1} [x(n) + x(n+N/4) \\
&\quad + x(n+N/2) + x(n+3N/4)]\, W_N^0\, W_{N/4}^{nr} \\
X(4r+1) &= \sum_{n=0}^{(N/4)-1} [x(n) - j\, x(n+N/4) \\
&\quad - x(n+N/2) + j\, x(n+3N/4)]\, W_N^n\, W_{N/4}^{nr} \\
X(4r+2) &= \sum_{n=0}^{(N/4)-1} [x(n) - x(n+N/4) \\
&\quad + x(n+N/2) - x(n+3N/4)]\, W_N^{2n}\, W_{N/4}^{nr} \\
X(4r+3) &= \sum_{n=0}^{(N/4)-1} [x(n) + j\, x(n+N/4) \\
&\quad - x(n+N/2) - j\, x(n+3N/4)]\, W_N^{3n}\, W_{N/4}^{nr}
\end{aligned} \tag{19}
$$

Each one of these equations is now an N/4-point FFT that can be computed by repeating the above procedure until N=4. Note that the factors W_N^0, W_N^n, W_N^{2n}, and W_N^{3n} are considered part of the signal. In general, an N-point FFT (where N is a power of 4) can be reduced to the computation of four N/4-point FFTs by transforming the input signal x(n) into an intermediate signal y(n), as suggested by (19). Figure 17 shows the corresponding radix-4 DIF butterfly, which generates one term for each sum in (19).

For simplicity, the notation of Figure 18 is often used instead of that of Figure 17 for the butterfly of radix-4 DIF FFT.

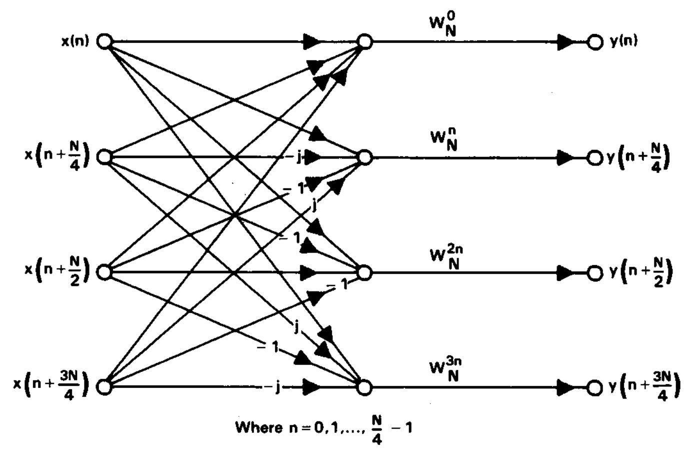

Figure 17. Radix-4 DIF Butterfly

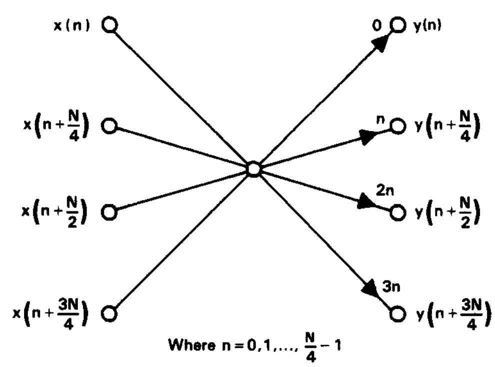

Figure 18. Alternate Form of the Radix-4 DIF Butterfly

Figure 19 shows an example of a 64-point, radix-4 DIF FFT.

Note that the inputs are normally ordered while the outputs are presented in a digit-reversed order. The principle of digit reversal is the same as in radix-2 FFT, but now the digits are 0, 1, 2, and 3 (quaternary system) instead of 0 and 1 (binary system). The code for digit reversal is the same as that shown in Figure 11. For example, the datapoint occupying location 132 (quaternary number corresponding to decimal 30) exchanges positions with the datapoint at location 231 (corresponding to the decimal 45).

Another important point of the radix-4 algorithm regards scaling. Since each stage of the radix-4 algorithm corresponds to two stages of the radix-2 algorithm, equivalent results are obtained by dividing the output of each stage of the radix-4 algorithm by 4.

Appendix E contains the implementation of a 256-point, radix-4 DIF FFT on the TMS32020. This implementation follows the one described in FORTRAN code in the book by Burrus and Parks.[4]

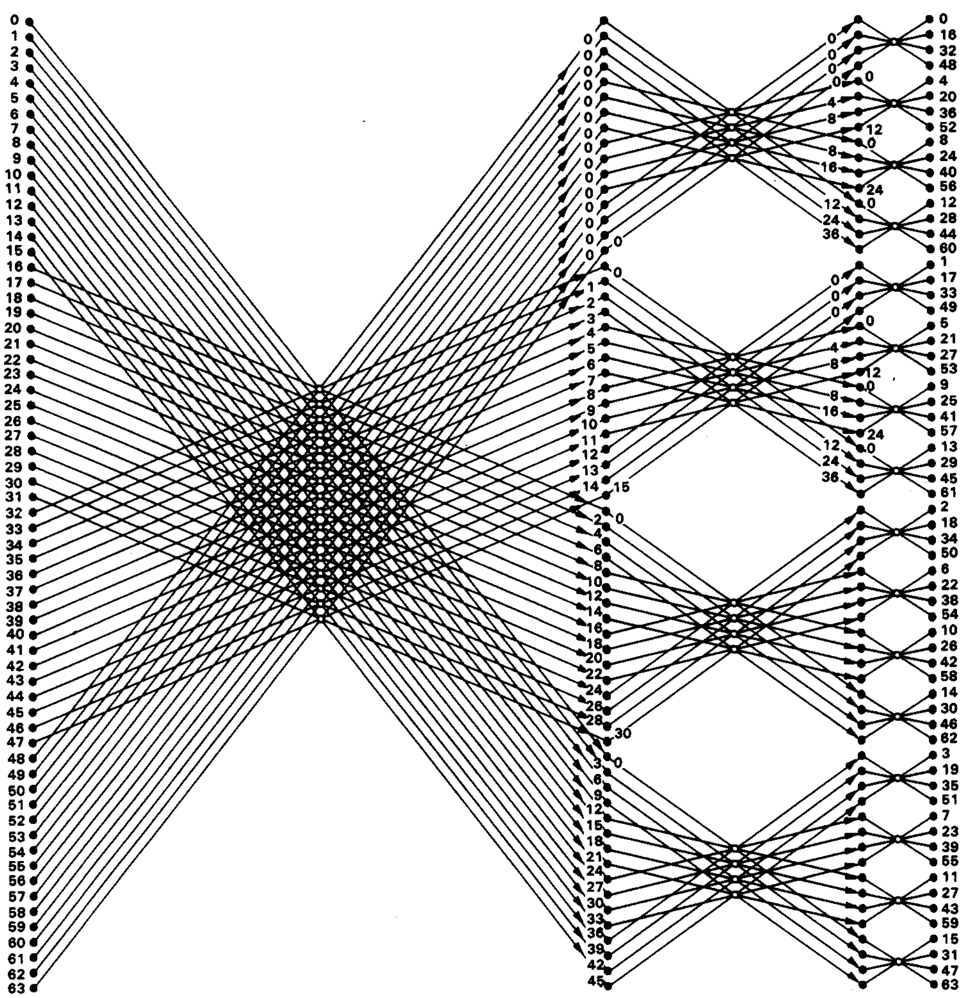

Figure 19. A 64-Point, Radix-4 DIF FFT

SYSTEM MEMORY AND I/O CONSIDERATIONS

Unlike non-realtime FFT applications where data samples to be transformed are assumed to already be in data memory, realtime FFT applications demand careful considerations of data input/output and system memory utilization.

The TMS32020 has 544 words of on-chip data RAM, organized into two 256-word blocks (B0 and B1) and one 32-word block (B2) that can be used as scratch-pad locations.[5] In non-realtime applications, this memory configuration allows a 256-point complex FFT to be easily performed (see Appendix C). However, for realtime FFT applications, input/output data buffering is generally required.

For small transform sizes, up to 128-point complex (or 256-point real) FFTs, the double-buffering technique, shown in Figure 20, can be used for realtime applications without the need of any external data memory. The on-chip RAM blocks B0 and B1 are organized into Buffer A and Buffer B, respectively.

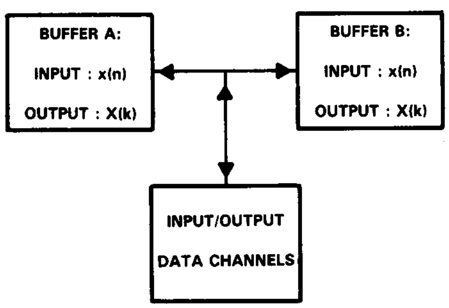

Figure 20. Input/Output Double-Buffering

Consider a 128-point complex FFT. Realtime input data to be transformed can be grouped into "frames" of 256 words (128 complex inputs) read into either Buffer A or Buffer B, depending on which one is not currently being used by the FFT program. The idea is to use the two on-chip RAM blocks B0 and B1 alternatively as I/O and transform buffers.

Assuming that the frame of data in Buffer A, the current transform buffer, is being transformed in-place, a software flag is then set to indicate that Buffer B can now be used as the current I/O buffer. This means that while time-domain data is read into Buffer B, the current I/O buffer, previous transformed data in Buffer B must be transferred out at the same time to make room for the incoming data. This can be acomplished efficiently if the I/O transfers are sequential and organized in a back-to-back manner (i.e., an output operation followed by an input operation).

Resetting the flag indicates that the roles of Buffer A and Buffer B are now reversed. In this case, Buffer B now has a full frame of input data ready to be transformed while Buffer A has a full frame of transformed data (spectral samples) ready to be transferred out to make room for more incoming time-domain data. The setting of the software flag is often implemented as an I/O device service routine (DSR) or as an interrupt handler in the case of interrupt-driven I/O.

Although this double-buffering technique is also applicable to larger transforms with the use of external memory, the actual memory required can be optimized if the transform time for an N-point FFT is shorter than the time to assemble a frame of N complex input data samples. For this purpose, the circular-buffer technique, shown in Figure 21, can be used.

Instead of a double-buffer size of 2N, a circular-buffer size of $N+M < 2N$ can be used where $M < N$ and M depends on the system input data rate in general. For example, M is chosen to be no less than 8T for an 8-kHz input sampling frequency and an N-point complex FFT with a transform time of T ms.

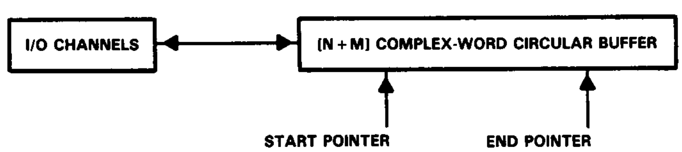

Figure 21. Input/Output Circular-Buffering for Large FFTs

A set of pointers is used to manage the data in the circular buffer. The start pointer is set at the beginning of the current frame, whereas the end pointer always indicates the current input data position in the circular buffer. Both pointers "wrap around" at the end of the circular buffer.

When a complete frame of input data has been collected, the set of pointer values is passed to the FFT program to transform the frame of data. For the next frame of input data, the start pointer points to the location immediately following the last location of the previous frame. As before, the end pointer for the current frame tracks the location of the next input data, and the whole process is repeated.

To decrease execution time, a large N-point FFT can be divided into smaller 256-point complex FFTs and executed 256 complex points at a time utilizing the on-chip RAM, as shown in Figure 22. Note that the system is still collecting incoming time-domain data samples and storing them in the external circular buffer while the FFT program is executing with internal data RAM. When 256 complex points have been processed, the FFT program returns them to the external buffer while fetching the next set of 256 samples for execution.

This scheme takes advantage of the fact that off-chip data accesses take two cycles each while on-chip data accesses take one cycle each. Certain instructions (e.g., SACL and SACH) even take three cycles to execute when operating on external RAM. To speed execution, off-chip data blocks can be efficiently moved into on-chip data memory via the BLKD (block move from data memory to data memory) instruction, which executes in a single cycle when used in the repeat mode with the repeat counter having a maximum count of 256.

IMPLEMENTING LARGE FFT'S

Figure 23 shows the memory configurations and transfers for a 1024-point complex FFT computed as four 256-point complex FFTs. A kernel 256-point complex FFT can operate on a group of 256 complex points at one time using on-chip RAM. Data transfers between on-chip and off-chip RAM are efficiently performed via the RPTK and BLKD instructions.

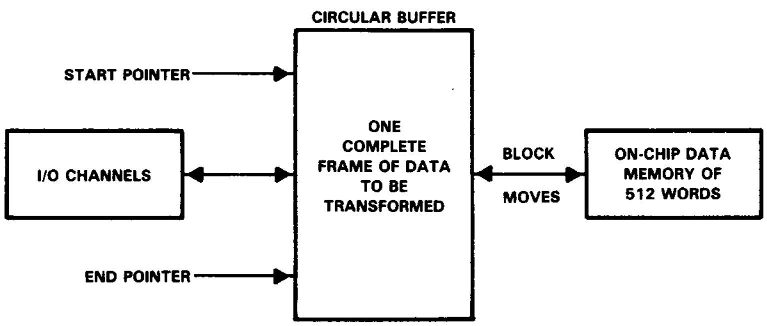

Figure 22. Use of On-Chip Memory to Speed FFT Execution

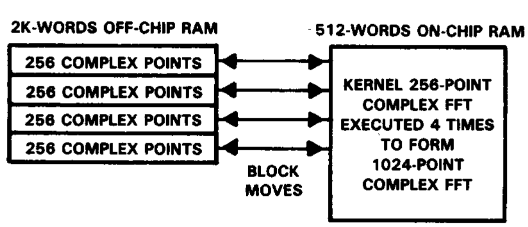

Figure 23. Execution of a 1024-Point Complex FFT with On-Chip RAM

Figure 24 shows a more detailed block diagram of a 1024-point radix-2 complex FFT. It can be seen that 512 butterflies must be performed at each stage. The first eight stages have a total of 4096 butterflies computed by four 256-point FFTs. The 256-point FFT in Appendix C is used as a subroutine for this purpose. Appendix D contains a listing for the 1024-point complex FFT performed with the help of on-chip RAM. However, due to the size of the 1024-point FFT program, the user may find it necessary to subdivide the code into smaller sections prior to assembly.

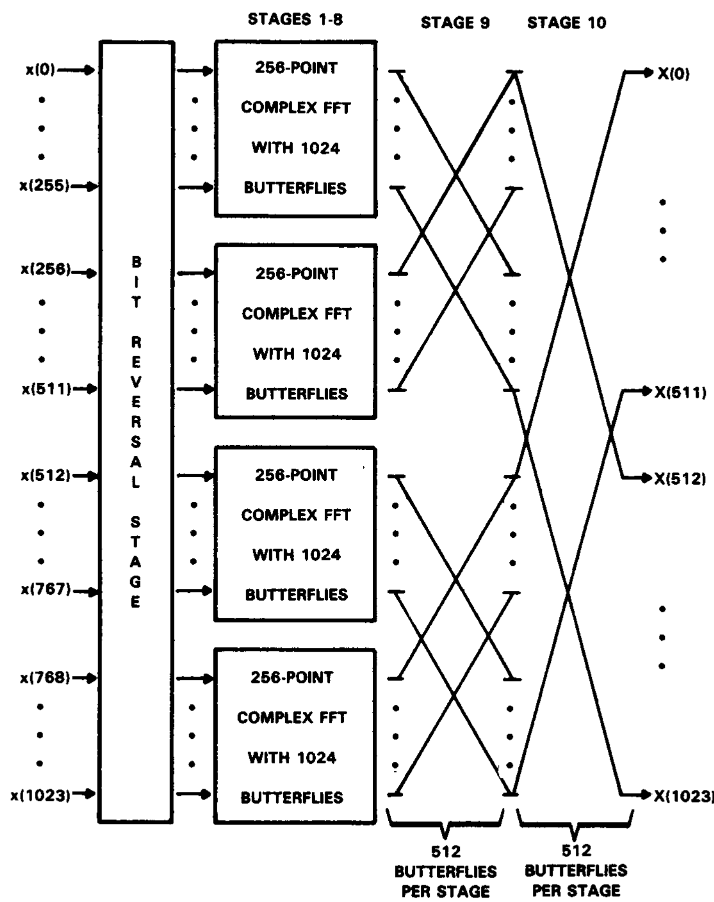

Figure 24. A 1024-Point Complex FFT Using a 256-Point Kernel for Stages 1-8

HIGHER-RADIX FFT'S

The same decomposition principle as for radix-2 FFT algorithms applies to higher radices as well. Table 3 shows the computation requirements[7] for a 4096-point FFT using various radices that are a power of two.

The main benefit of using higher-radix algorithms is the reduced amount of arithmetic operations required for the FFT computation. Beyond the use of radix-8 algorithms, however, the point of diminishing returns rapidly approaches. Memory addressing, data scaling, and program control become more and more complicated. On the other hand, a suitable combination of the radix-2, radix-4, and radix-8 algorithms becomes a flexible and efficient "mixed-radix" algorithm for most FFT applications. Reference [4] contains some useful FORTRAN routines for higher-radix FFT algorithms.

REAL TRANSFORMS VIA COMPLEX FFT'S

In practice, many signals are real functions of time, whereas the FFT algorithm has been derived for complex signals. This means that for real inputs, the imaginary parts of the complex entries are simply set to zero. This results in a certain amount of redundancy. To utilize the bandwidth of the FFT algorithm more effectively, one can use the fact that the frequency spectrum of a real signal is a hermitian function (i.e., the real part is an even function while the imaginary part is an odd function).[11] For example, two N-point real FFTs can be computed simultaneously with a single N-point complex FFT.[8,9,10] On the other hand, a single N-point complex FFT can also be utilized effectively to perform a 2N-point real FFT.[10] Such algorithms substantially reduce the amount of computation required for real FFTs.

Other efficient algorithms exist that further utilize the properties of real signals. In particular, the FFTs of four N-point real symmetric (even) and antisymmetric (odd) sequences can be computed with just one N-point complex FFT.[2] Alternatively, the same N-point complex FFT can be used to compute the FFTs of four N-point real symmetric (even) sequences simultaneously.[2]

INVERSE FFT

The inverse FFT is given by the equation

$$x(n) = (1/N) \sum_{k=0}^{N-1} X(k)\, W_N^{-nk} \qquad n=0,1,\ldots,N-1 \qquad (20)$$

where X(k) is the Fourier transform of the time-domain signal x(n). Note that (20) is essentially the same equation as (1), which represents the "forward" FFT, with two important differences: the scaling factor is (1/N), and the exponent of the twiddle factor is the negative of the one in equation (1). Because of the similarity between (20) and (1), the implementation of the inverse FFT is very straightforward. The code can be derived from that given in the appendices by applying the following modifications.

If the forward FFT is implemented with scaling, the resulting values in the frequency domain are (1/N)X(k) and not X(k). Hence, for the inverse FFT, no scaling must be applied in order to get back the original signal. On the other hand, if the forward FFT has not been scaled, the inverse FFT must be scaled. This scaling can be performed all at one point as suggested by (20), or at every stage as described earlier in the forward FFT.

The negative exponent of the twiddle factors implies that the values of sin(X) will have the opposite sign from that in the forward FFT. Therefore, one way to implement the inverse FFT is to have an additional table with the negatives of sin(X). Another method is possible if the complex conjugate of (20) is considered.

$$\overset{*}{x}(n) = (1/N) \sum_{k=0}^{N-1} \overset{*}{X}(k)\, W_N^{nk} \qquad n=0,1,\ldots,N-1 \qquad (21)$$

In (21), the asterisk indicates complex conjugate. In this form, there is no need to have an additional table for sin(X). Instead, the inverse FFT is implemented by applying the forward FFT on the complex conjugate of X(k) (with appropriate scaling). The complex conjugate of the resulting time signal is the desired result. Note that if x(n) is real, this last step is not necessary since, in this case, $\overset{*}{x}(n) = x(n)$.

Table 3. Computational Requirements for Higher-Radix FFT Algorithms

ALGORITHM	NUMBER OF REAL MULTIPLICATIONS	NUMBER OF REAL ADDITIONS
RADIX-2 $(N=2^{12})$	81,924	139,266
RADIX-4 $(N=4^{6})$	57,348	126,978
RADIX-8 $(N=8^{4})$	49,156	126,978
RADIX-16 $(N=16^{3})$	48,132	125,442

Table 4. FFT Performance for a TMS32020 Implementation

FFT			EXECUTION		
ALGORITHM	SIZE	TYPE	CYCLES	CLOCK	TIME
RADIX-2	128-Pt	Looped	21,879	5 MHz	4.375 ms
RADIX-2	256-Pt	Looped	42,416	5 MHz	8.483 ms
RADIX-2	256-Pt	Straight-Line	22,595	5 MHz	4.519 ms
RADIX-2	1024-Pt	Straight-Line	159,099	5 MHz	31.8198 ms
RADIX-4	256-Pt	Straight-Line	15,551	5 MHz	3.1102 ms

FFT PERFORMANCE TIMING

Table 4 provides the FFT timing performance for the TMS32020 code in the appendices. The source code examples included in Appendices C through G are not optimized for any specific application since they have been designed to emphasize clarity rather than code optimization. The key feature of these codes is that they do not require any scratch-pad (temporary) memory locations. Consequently, these codes should be useful in memory-critical applications. For time-critical applications, the codes can be optimized for better execution time. Higher execution speed is achieved by using straightline instead of looped code. The tradeoff for this optimization is the larger program memory requirements of the straightline code.

The TMS32020 digital signal processor offers many advantages for the implementation of FFT algorithms. Its 200-ns cycle time and special features, such as the single-cycle multiplication, allow high execution speed. The 544 16-bit words of on-chip memory permit the implementation of a 256-point complex FFT without access to external memory, thus further reducing execution time. Furthermore, special instructions, such as RPTK and BLKD, allow the quick transfer of data from external to internal memory, so that portions of large FFTs can be implemented with the on-chip RAM. Due to the flexibility of the TMS32020, the designer can trade-off program memory with execution speed.

DESCRIPTION OF THE APPENDICES

At the end of this report, there are five appendices with TMS32020 code implementing several FFTs. The contents of the appendices are the following:

Appendix A: N-point, radix-2, DIT FFT (9 macros)
Appendix B: N-pôint, radix-2, DIT FFT using indirect addressing (7 macros)
Appendix C: 256-point, radix-2 DIT FFT
Appendix D: 1024-point, radix-2 DIT FFT
Appendix E: 256-point, radix-4 DIF FFT
Appendix F: 128-point, radix-2 DIF FFT (looped code)
Appendix G: 256-point, radix-2 DIF FFT (looped code)

SUMMARY

The purpose of this report has been to develop an understanding of the underlying principles in FFT implementations with the TMS32020 processor. The book by Burrus and Parks[4] contains examples of FFT implementations on the TMS32010 processor, the first member of the TMS320 family.

This report has discussed the development of the DFT algorithm, leading to the derivation of the FFT algorithm. The implementation of the radix-2 DIT FFT algorithm was covered in detail, and the radix-4 DIF FFT algorithm was also explained. Special attention was given to various FFT implementation aspects, such as scaling, system memory, and input/outpput considerations.

REFERENCES

1. B. Gold and C.M. Rader, *Digital Processing of Signals*, McGraw-Hill (1969).
2. A.V. Oppenheim and R.W. Schafer, *Digital Signal Processing*, Prentice-Hall (1975).
3. L.R. Rabiner and B. Gold, *Theory and Applications of Digital Signal Processing*, Prentice-Hall (1975).
4. C.S. Burrus and T.W. Parks, *DFT/FFT and Convolution Algorithms - Theory and Implementation*, John Wiley & Sons (1985).
5. *TMS32020 User's Guide*, Texas Instruments (1985).
6. H.W. Johnson and C.S. Burrus, "An In-order, In-place Radix-2 FFT," *1984 IEEE ICASSP Proceedings*, 28A.2.1-2.4 (March 1984).
7. G.D. Bergland, "A Fast Fourier Transform Algorithm using Base-8 Iterations," *Mathematics of Computation*, **Vol 22**, No. 102, 275-279 (April 1968).
8. J.W. Cooley, P.A.W. Lewis, and P.D. Welch, "The Fast Fourier Transform Algorithm - Programming Considerations in the Calculation of Sine, Cosine, and Laplace Transforms," *Journal of Sound Vibration*, **Vol 12**, 315-337 (July 1970).
9. G.D. Bergland, "A Radix-8 Fast Fourier Transform Subroutine for Real Valued Series," *IEEE Transcriptions of Audio and Electroactivity*, **Vol AU-17**, No. 2 (June 1969).
10. E.O. Brigham, *The Fourier Transform*, Prentice-Hall (1974).
11. R Bracewell, *The Fourier Transform and Its Applications*, McGraw-Hill (1965).

APPENDIX A
FFT MACRO LIBRARY (DIRECT ADDRESSING)

APPENDIX A

```
****************************************************************
*                                                              *
*  THIS TMS32020 FFT MACRO LIBRARY CONTAINS A TOTAL OF 9       *
*  MACROS FOR IMPLEMENTING A GENERAL RADIX-2 DIT N-POINT       *
*  FFT.  ALL DIT BUTTERFLIES ARE IMPLEMENTED WITH DYNAMIC      *
*  SCALING TO AVOID ARITHMETIC OVERFLOWS. TWO'S-COMPLEMENT     *
*  FIXED-POINT FRACTIONAL ARITHMETIC IS USED THROUGHOUT.       *
*  WHILE THESE MACROS ARE NOT NECESSARILY OPTIMIZED FOR        *
*  SPEED, THEY ARE SO STRUCTURED THAT NO INTERMEDIATE OR       *
*  TEMPORARY REGISTERS ARE REQUIRED FOR THEIR EXECUTION.       *
*  CONSEQUENTLY, THESE MACROS ARE PARTICULARLY USEFUL IN       *
*  APPLICATIONS WHERE MEMORY SPACE PROVES TO BE CRITICAL.      *
*  IN ADDITION,  ALL FFT COMPUTATIONS ARE PERFORMED            *
*  IN-PLACE AND THE REAL AND IMAGINARY PARTS OF ALL COMPLEX*
*  INPUTS ARE ASSUMED TO BE IN CONSECUTIVE DATA MEMORY         *
*  LOCATIONS.                                                  *
*                                                              *
*                                                              *
****************************************************************
*                                                              *
****************************************************************
*                                                              *
*                                         K                    *
*          P .                   . P = P + Q*W                 *
*              .               '          N                    *
*                .           '                                 *
*                  .       '                                   *
*                    .   '                                     *
*                      O                                       *
*                    '   .                                     *
*                  '       .                                   *
*                ' K         .                                 *
*              '  W            .          K                    *
*          Q .    N              . Q = P - Q*W                 *
*                                         N                    *
*                                                              *
*                                                              *
*     NOTATION FOR IN-PLACE RADIX-2 DIT FFT BUTTERFLY          *
*                                                              *
****************************************************************
*                                                              *
****************************************************************
*                                                              *
*                                                            M *
* IN A RADIX-2 N-POINT FFT, THERE ARE M PASSES WHERE N =2      *
* WITH [N/2] 2-POINT BUTTERFLIES PER PASS, GIVING A TOTAL      *
* OF [N/2]LOG N BUTTERFLIES PER FFT.                           *
*            2                                                 *
*                                                              *
****************************************************************
```

```
****************************************************************
*                                                              *
*       MACRO 1: W =1, k=0.                                    *
*                 N                                            *
*                                                              *
*       P=(PR+jPI)      P+Q=(PR+QR)+j(PI+QI)                   *
*                    \ /                                       *
*                    / \                                       *
*       Q=(QR+jQI)      P-Q=(PR-QR)+j(PI-QI)                   *
*                                                              *
*                                                              *
*          -j(2(pi)/N)k                                        *
*       W =e            =COS[(2(pi)/N)k]-jSIN[(2(pi)/N)k]      *
*        N                                                     *
*                      =WR+jWI                                 *
*                                                              *
*                      =1                                      *
*                                                              *
*                                                              *
*  ALL OUTPUT SAMPLES ARE SCALED DOWN BY 2 TO ACCOMMODATE      *
*  A 1-BIT OVERFLOW.  HOWEVER, NO OVERFLOWS WILL OCCUR         *
*  FOR FRACTIONAL INPUTS OF THE FORM X : -1 <= X < 1.  A       *
*  TOTAL OF 10 INSTRUCTIONS IS USED.  EXECUTION                *
*  TIME IS EQUAL TO 10 MACHINE CYCLES.                         *
*                                                              *
****************************************************************
ZERO     $MACRO     PR,PI,QR,QI
*
*        CALCULATE Re[P+Q] AND Re[P-Q]
*
         LAC        :PR:,15   ACC := (1/2)(PR)
         ADD        :QR:,15   ACC := (1/2)(PR+QR)
         SACH       :PR:      PR  := (1/2)(PR+QR)
         SUBH       :QR:      ACC := (1/2)(PR+QR)-(QR)
         SACH       :QR:      QR  := (1/2)(PR-QR)
*
*        CALCULATE Im[P+Q] AND Im[P-Q]
*
         LAC        :PI:,15   ACC := (1/2)(PI)
         ADD        :QI:,15   ACC := (1/2)(PI+QI)
         SACH       :PI:      PI  := (1/2)(PI+QI)
         SUBH       :QI:      ACC := (1/2)(PI+QI)-(QI)
         SACH       :QI:      QI  := (1/2)(PI-QI)
         $END
*
```

```
*************************************************************
*                                                           *
*       MACRO 2: W =-1, k=N/2.                              *
*                 N                                         *
*                                                           *
*       P=(PR+jPI)      P-Q=(PR-QR)+j(PI-QI)                *
*                   \ /                                     *
*                   / \                                     *
*       Q=(QR+jQI)      P+Q=(PR+QR)+j(PI+QI)                *
*                                                           *
*                                                           *
*           -j(2(pi)/N)k                                    *
*       W =e              =COS[(2(pi)/N)k]-jSIN[(2(pi)/N)k] *
*        N                                                  *
*                         =WR+jWI                           *
*                                                           *
*                         =-1                               *
*                                                           *
*                                                           *
*  ALL OUTPUT SAMPLES ARE SCALED DOWN BY 2 TO ACCOMMODATE   *
*  A 1-BIT OVERFLOW.  HOWEVER, NO OVERFLOWS WILL OCCUR      *
*  FOR FRACTIONAL INPUTS OF THE FORM X : -1 <= X < 1.  A    *
*  TOTAL OF 10 INSTRUCTIONS IS USED.  EXECUTION             *
*  TIME IS EQUAL TO 10 MACHINE CYCLES.                      *
*                                                           *
*************************************************************
PI      $MACRO    PR,PI,QR,QI
*
*       CALCULATE Re[P+Q] AND Re[P-Q]
*
        LAC       :PR:,15   ACC := (1/2)(PR)
        SUB       :QR:,15   ACC := (1/2)(PR-QR)
        SACH      :PR:      PR  := (1/2)(PR-QR)
        ADDH      :QR:      ACC := (1/2)(PR-QR)+(QR)
        SACH      :QR:      QR  := (1/2)(PR+QR)
*
*       CALCULATE Im[P+Q] AND Im[P-Q]
*
        LAC       :PI:,15   ACC := (1/2)(PI)
        SUB       :QI:,15   ACC := (1/2)(PI-QI)
        SACH      :PI:      PI  := (1/2)(PI-QI)
        ADDH      :QI:      ACC := (1/2)(PI-QI)+(QI)
        SACH      :QI:      QI  := (1/2)(PI+QI)
        $END
*
```

```
*************************************************************
*                                                           *
*       MACRO 3: W =-j, k=N/4.                              *
*                 N                                         *
*                                                           *
*       P=(PR+jPI)      P-jQ=(PR+QI)+j(PI-QR)               *
*                   \ /                                     *
*                   / \                                     *
*       Q=(QR+jQI)      P+jQ=(PR-QI)+j(PI+QR)               *
*                                                           *
*                                                           *
*           -j(2(pi)/N)k                                    *
*       W =e              =COS[(2(pi)/N)k]-jSIN[(2(pi)/N)k] *
*        N                                                  *
*                         =WR+jWI                           *
*                                                           *
*                         =-j                               *
*                                                           *
*                                                           *
*  ALL OUTPUT SAMPLES ARE SCALED DOWN BY 2 TO ACCOMMODATE   *
*  A 1-BIT OVERFLOW.  HOWEVER, NO OVERFLOWS WILL OCCUR      *
*  FOR FRACTIONAL INPUTS OF THE FORM X : -1 <= X < 1.  A    *
*  TOTAL OF 11 INSTRUCTIONS IS USED.  EXECUTION             *
*  TIME IS EQUAL TO 11 MACHINE CYCLES.                      *
*                                                           *
*  WARNING: THIS MACRO REQUIRES THE INPUT SAMPLES QR AND    *
*           QI TO BE IN CONSECUTIVE DATA MEMORY LOCATIONS   *
*           OF ASCENDING ORDER. THE FOLLOWING STEPS ARE     *
*           USED TO IMPLEMENT THIS MACRO:                   *
*                                                           *
*         (1) [PI-QR] -----> [PI]                           *
*         (2) [PI+QR] -----> [QR]                           *
*         (3) [PR+QI] -----> [PR]                           *
*         (4) [PR-QI] -----> [ACC]                          *
*         (5) [QR]    -----> [QI]                           *
*         (6) [ACC]   -----> [QR]                           *
*                                                           *
*************************************************************
PIBY2   $MACRO    PR,PI,QR,QI
*
*       CALCULATE Re[P+jQ] AND Re[P-jQ]
*
        LAC       :PI:,15   ACC := (1/2)(PI)
        SUB       :QR:,15   ACC := (1/2)(PI-QR)
        SACH      :PI:      PI  := (1/2)(PI-QR)
        ADDH      :QR:      ACC := (1/2)(PI-QR)+(QR)
        SACH      :QR:      QR  := (1/2)(PI+QR)
*
*       CALCULATE Im[P+jQ] AND Im[P-jQ]
*
        LAC       :PR:,15   ACC := (1/2)(PR)
        ADD       :QI:,15   ACC := (1/2)(PR+QI)
```

```
        SACH     :PR:       PR  := (1/2)(PR+QI)
        SUBH     :QI:       ACC := (1/2)(PR+QI)-(QI)
        DMOV     :QR:       QR  -> QI
        SACH     :QR:       QR  := (1/2)(PR-QI)
        $END
*


*************************************************************
*                                                           *
*       MACRO 4: W =j, k=3N/4.                              *
*                 N                                         *
*                                                           *
*       P=(PR+jPI)     P+jQ=(PR-QI)+j(PI+QR)                *
*                  \ /                                      *
*                  / \                                      *
*       Q=(QR+jQI)     P-jQ=(PR+QI)+j(PI-QR)                *
*                                                           *
*                                                           *
*           -j(2(pi)/N)k                                    *
*       W =e            =COS[(2(pi)/N)k]-jSIN[(2(pi)/N)k]   *
*        N                                                  *
*                      =WR+jWI                              *
*                                                           *
*                      =j                                   *
*                                                           *
*                                                           *
*  ALL OUTPUT SAMPLES ARE SCALED DOWN BY 2 TO ACCOMMODATE   *
*  A 1-BIT OVERFLOW.  HOWEVER, NO OVERFLOWS WILL OCCUR      *
*  FOR FRACTIONAL INPUTS OF THE FORM X : -1 <= X < 1.  A    *
*  TOTAL OF 10 INSTRUCTIONS ARE USED SUCH  THAT EXECUTION   *
*  TIME IS EQUAL TO 10 MACHINE CYCLES.                      *
*                                                           *
*  WARNING: THIS MACRO REQUIRES THE INPUT SAMPLES QR AND    *
*           QI TO BE IN CONSECUTIVE DATA MEMORY LOCATIONS   *
*           ASCENDING ORDER. THE FOLLOWING STEPS ARE USED   *
*           TO IMPLEMENT THIS MACRO:                        *
*                                                           *
*           (1) [PI-QR] -----> [PI]                         *
*           (2) [PI+QR] -----> [QR]                         *
*           (3) [PR+QI] -----> [PR]                         *
*           (4) [PR-QI] -----> [ACC]                        *
*           (5) [QR]    -----> [QI]                         *
*           (6) [ACC]   -----> [QR]                         *
*                                                           *
*************************************************************
PI3BY2  $MACRO    PR,PI,QR,QI
*
*       CALCULATE Re[P+jQ] AND Re[P-jQ]
*
        LAC      :PI:,15    ACC := (1/2)(PI)
        ADD      :QR:,15    ACC := (1/2)(PI+QR)
        SACH     :PI:       PI  := (1/2)(PI+QR)
        SUBH     :QR:       ACC := (1/2)(PI+QR)-(QR)
        SACH     :QR:       QR  := (1/2)(PI-QR)
*
```

```
*       CALCULATE Im[P+jQ] AND Im[P-jQ]
*
        LAC      :PR:,15    ACC := (1/2)(PR)
        SUB      :QI:,15    ACC := (1/2)(PR-QI)
        SACH     :PR:       PR  := (1/2)(PR-QI)
        ADDH     :QI:       ACC := (1/2)(PR-QI)+(QI)
        DMOV     :QR:       QR  -> QI
        SACH     :QR:       QR  := (1/2)(PR+QI)
        $END
*


*************************************************************
*                                                           *
*       MACRO 5:     k=N/8.                                 *
*                                                           *
*       P=(PR+jPI)   P+Q*W=(PR+Re[Q*W])+j(PI+Im[Q*W])       *
*                 \ /                                       *
*                 / \                                       *
*       Q=(QR+jQI)   P-Q*W=(PR-Re[Q*W])+j(PI-Im[Q*W])       *
*                                                           *
*           -j(2(pi)/N)k                                    *
*       W =e            =COS((pi)/4)-jSIN((pi)/4)=WR+jWI    *
*        N                                                  *
*                                                           *
*       LET          W=|COS((pi)/4)|=|SIN((pi)/4)|          *
*                                                           *
*       THEN      [Q*W]=(QR+QI)*W+j(QI-QR)*W                *
*                                                           *
*              Re[Q*W]=(QI+QR)*W                            *
*                                                           *
*              Im[Q*W]=(QI-QR)*W                            *
*                                                           *
*                                                           *
*  ALL OUTPUT SAMPLES ARE SCALED DOWN BY 2 TO ACCOMMODATE   *
*  A 1-BIT OVERFLOW.  HOWEVER, NO OVERFLOWS WILL OCCUR      *
*  FOR FRACTIONAL INPUTS OF THE FORM X : -1 <= X < 1.  A    *
*  TOTAL OF 20 INSTRUCTIONS ARE USED SUCH  THAT EXECUTION   *
*  TIME IS EQUAL TO 20 MACHINE CYCLES.   THIS MACRO         *
*  REQUIRES W TO BE THE ABSOLUTE VALUE (MAGNITUDE)  OF      *
*  COS((pi)/4) AND SIN((pi)/4).  THE SIGNS OF THESE TRIG    *
*  FUNCTIONS HAVE BEEN TAKEN CARE OF IN THE CODE.           *
*                                                           *
*************************************************************
PIBY4   $MACRO    PR,PI,QR,QI,W
*
        LT       :W:        T-REGISTER :=W=COS(PI/4)=SIN(PI/4)
        LAC      :QI:,14    ACC := (1/4)(QI)
        SUB      :QR:,14    ACC := (1/4)(QI-QR)
        SACH     :QI:,1     QI  := (1/2)(QI-QR)
        ADD      :QR:,15    ACC := (1/4)(QI+QR)
        SACH     :QR:,1     QR  := (1/2)(QI+QR)
        LAC      :PR:,14    ACC := (1/4)(PR)
        MPY      :QR:       P-REGISTER := (1/4)(QI+QR)*W
        APAC                ACC := (1/4)[PR+(QI+QR)*W]
        SACH     :PR:,1     PR  := (1/2)[PR+(QI+QR)*W]
```

```
         SPAC                ACC := (1/4)(PR)
         SPAC                ACC := (1/4)[PR-(QI+QR)*W]
         SACH      :QR:,1    QR  := (1/2)[PR-(QI+QR)*W]
         LAC       :PI:,14   ACC := (1/4)(PI)
         MPY       :QI:      P-REGISTER := (1/4)(QI-QR)*W
         APAC                ACC := (1/4)[PI+(QI-QR)*W]
         SACH      :PI:,1    PI  := (1/2)[PI+(QI-QR)*W]
         SPAC                ACC := (1/4)(PI)
         SPAC                ACC := (1/4)[PI-(QI-QR)*W]
         SACH      :QI:,1    QI  := (1/2)[PI-(QI-QR)*W]
         $END
*

**************************************************************
*                                                            *
*        MACRO 6:     k=3N/8.                                *
*                                                            *
*        P=(XR+jXI)   X+Q*W=(XR+Re[Q*W])+j(XI-Im[Q*W])       *
*                  \ /                                       *
*                  / \                                       *
*        Q=(QR+jQI)   X-Q*W=(XR-Re[Q*W])+j(XI+Im[Q*W])       *
*                                                            *
*            -j(2(pi)/N)k                                    *
*        W =e            =COS(3(pi)/4)-jSIN(3(pi)/4)         *
*         N                                                  *
*                      =WR+jWI                               *
*                                                            *
*        LET          W=|COS(3(pi)/4)|=|SIN(3(pi)/4)|        *
*                                                            *
*        THEN     [Q*W]=(QI-QR)*W-j(QI+QR)*W                 *
*                                                            *
*               RE[Q*W]=(QI-QR)*W                            *
*                                                            *
*               IM[Q*W]=(QI+QR)*W                            *
*                                                            *
*                                                            *
*  ALL OUTPUT SAMPLES ARE SCALED DOWN BY 2 TO ACCOMMODATE    *
*  A 1-BIT OVERFLOW.  HOWEVER, NO OVERFLOWS WILL OCCUR       *
*  FOR FRACTIONAL INPUTS OF THE FORM X : -1 <= X < 1.  A     *
*  TOTAL OF 20 INSTRUCTIONS ARE USED SUCH  THAT EXECUTION    *
*  TIME IS EQUAL TO 20 MACHINE CYCLES.   THIS MACRO          *
*  REQUIRES W TO BE THE ABSOLUTE VALUE (MAGNITUDE)  OF       *
*  COS(3(pi)/4) AND SIN(3(pi)/4).  THE SIGNS OF THESE TRIG   *
*  FUNCTIONS HAVE BEEN TAKEN CARE OF IN THE CODE.            *
*                                                            *
**************************************************************
PI3BY4   $MACRO    PR,PI,QR,QI,W
*
         LT        :W:       T-REGISTER :=W=COS(PI/4)=SIN(PI/4)
         LAC       :QI:,14   ACC := (1/4)(QI)
         SUB       :QR:,14   ACC := (1/4)(QI-QR)
         SACH      :QI:,1    QI  := (1/2)(QI-QR)
         ADD       :QR:,15   ACC := (1/4)(QI+QR)
```

```
         SACH      :QR:,1    QR  := (1/2)(QI+QR)
         LAC       :PR:,14   ACC := (1/4)(PR)
         MPY       :QI:      P-REGISTER := (1/4)(QI-QR)*W
         APAC                ACC := (1/4)[PR+(QI-QR)*W]
         SACH      :PR:,1    PR  := (1/2)[PR+(QI-QR)*W]
         SPAC                ACC := (1/4)(PR)
         SPAC                ACC := (1/4)[PR-(QI-QR)*W]
         MPY       :QR:      P-REGISTER := (1/4)(QI+QR)*W
         SACH      :QR:,1    QR  := (1/2)[PR-(QI-QR)*W]
         LAC       :PI:,14   ACC := (1/4)(PI)
         SPAC                ACC := (1/4)[PI-(QI+QR)*W]
         SACH      :PI:,1    PI  := (1/2)[PI-(QI+QR)*W]
         APAC                ACC := (1/4)(PI)
         APAC                ACC := (1/4)[PI+(QI+QR)*W]
         SACH      :QI:,1    QI  := (1/2)[PI+(QI+QR)*W]
         $END
*

**************************************************************
*                                                            *
*        MACRO 7:  A GENERAL RADIX-2 DIT FFT 'BUTTERFLY'     *
*                                                            *
*        P=(PR+jPI)   P+Q*W=(PR+Re[Q*W])+j(PI-Im[Q*W])       *
*                  \ /                                       *
*                  / \                                       *
*        Q=(QR+jQI)   P-Q*W=(PR-Re[Q*W])+j(PI+Im[Q*W])       *
*                                                            *
*            -j(2(pi)/N)k                                      *
*        W =e            =COS(O)-jSIN(O)                     *
*         N                                                  *
*                        =WR+jWI                             *
*                                                            *
*                                                            *
*  ALL OUTPUT SAMPLES ARE SCALED DOWN BY 2 TO ACCOMMODATE    *
*  A 1-BIT OVERFLOW.  HOWEVER, NO OVERFLOWS WILL OCCUR       *
*  FOR FRACTIONAL INPUTS OF THE FORM X : -1 <= X < 1.  A     *
*  TOTAL OF 23 INSTRUCTIONS ARE USED SUCH  THAT EXECUTION    *
*  TIME IS EQUAL TO 23 MACHINE CYCLES.   THIS MACRO          *
*  REQUIRES W TO BE THE ABSOLUTE VALUE (MAGNITUDE)  OF       *
*  COS(O) AND SIN(O).  THE SIGNS OF THESE TRIG FUNCTIONS     *
*  HAVE BEEN TAKEN CARE OF IN THE CODE.                      *
*                                                            *
**************************************************************
NORM1    $MACRO         PR,PI,QR,QI,WR,WI
*
*        CALCULATE QR*WR + QI*WI AND STORE RESULT IN QI
*
         LT   :QI:      T-REGISTER := QI
         MPYK :WI:      P-REGISTER := (1/16)(QI*WI)
         LTP  :QR:      ACC := (1/16)(QI*WI); LOAD T WITH QR
         MPYK :WR:      P-REGISTER := (1/16)(QR*WR)
         APAC           ACC := (1/16)(QR*WR+QI*WI)
         SFR            ACC := (1/32)(QR*WR+QI*WI)
         SACH :QR:,4    QR  := (1/2)(QR*WR+QI*WI)
```

```
*
*       CALCULATE QI*WR - QR*WI AND STORE RESULT IN QR
*
        MPYK -:WI:      P-REGISTER := (1/16)(-QR*WI)
        LTP  :QI:       ACC := (1/16)(-QR*WI); T-REGISTER := QI
        MPYK :WR:       P-REGISTER := (1/16)(QI*WR)
        APAC            ACC := (1/16)(QI*WR-QR*WI)
        SFR             ACC := (1/32)(QI*WR-QR*WI)
        SACH :QI:,4     QI  := (1/2)(QI*WR-QR*WI)
*
*       CALCULATE Re[P+jQ] & Re[P-jQ] STORE RESULTS in PR & QR
*
        LAC  :PR:,14    ACC := (1/4)PR
        ADD  :QR:,15    ACC := (1/4)[PR+(QR*WR+QI*WI)]
        SACH :PR:,1     PR  := (1/2)[PR+(QR*WR+QI*WI)]
        SUBH :QR:       ACC := (1/4)[PR-(QR*WR+QI*WI)]
        SACH :QR:,1     QR  := (1/2)[PR-(QR*WR+QI*WI)]
*
*       CALCULATE Im[P+jQ] & Im[P-jQ] STORE RESULTS in PI & QI
*
        LAC  :PI:,14    ACC := (1/4)PI
        ADD  :QI:,15    ACC := (1/4)[PI+(QI*WR-QR*WI)]
        SACH :PI:,1     PR  := (1/2)[PI+(QI*WR-QR*WI)]
        SUBH :QI:       ACC := (1/4)[PI-(QI*WR-QR*WI)]
        SACH :QI:,1     QI  := (1/2)[PI-(QI*WR-QR*WI)]
        $END


*************************************************************
*                                                           *
*       MACRO 8:  A GENERAL RADIX-2 DIT FFT 'BUTTERFLY'     *
*                                                           *
*       P=(PR+jPI)    P+Q*W=(PR+Re[Q*W])+j(PI-Im[Q*W])      *
*                \ /                                        *
*                / \                                        *
*       Q=(QR+jQI)    P-Q*W=(PR-Re[Q*W])+j(PI+Im[Q*W])      *
*                                                           *
*            -j(2(pi)/N)k                                   *
*        W =e             =COS(O)-jSIN(O)                   *
*         N                                                 *
*                         =WR+jWI                           *
*                                                           *
*                                                           *
*  ALL OUTPUT SAMPLES ARE SCALED DOWN BY 2 TO ACCOMMODATE   *
*  A 1-BIT OVERFLOW.  HOWEVER, NO OVERFLOWS WILL OCCUR      *
*  FOR FRACTIONAL INPUTS OF THE FORM X : -1 <= X < 1.  A    *
*  TOTAL OF 22 INSTRUCTIONS ARE USED SUCH  THAT EXECUTION   *
*  TIME IS EQUAL TO 22 MACHINE CYCLES.   THIS MACRO         *
*  REQUIRES W TO BE THE ABSOLUTE VALUE (MAGNITUDE)  OF      *
*  COS(O) AND SIN(O).  THE SIGNS OF THESE TRIG FUNCTIONS    *
*  HAVE BEEN TAKEN CARE OF IN THE CODE.                     *
*                                                           *
*************************************************************
NORM2   $MACRO          PR,PI,QR,QI,WR,WI
*
*       CALCULATE QR*WR + QI*WI AND STORE RESULT IN QR
*
        LT   :QR:       T-REGISTER := QR
        MPY  :WR:       P-REGISTER := (1/2)(QR*WR)
        LTP  :QI:       ACC := (1/2)(QR*WR); T-REGISTER := QI
        MPY  :WI:       P-REGISTER := (1/2)(QI*WI)
        APAC            ACC := (1/2)(QR*WR+QI*WI)
        MPY  :WR:       P-REGISTER := (1/2)(QI*WR)
        LT   :QR:       LOAD T-REGISTER WITH QR
        SACH :QR:       QR  := (1/2)(QR*WR+QI*WI)
*
*       CALCULATE QI*WR - QR*WI AND STORE RESULT IN QI
*
        PAC             ACC= (1/2)(QI*WR)
        MPY  :WI:       P-REGISTER := (1/2)(QR*WI)
        SPAC            ACC= (1/2)(QI*WR-QR*WI)
        SACH :QI:       QI = (1/2)(QI*WR-QR*WI)
*
*       CALCULATE Re[Pm+1] & Re[Qm+1] STORE RESULTS in PR & QR
*
        LAC  :PR:,14    ACC= (1/4)PR
        ADD  :QR:,15    ACC= (1/4)[PR+(QR*WR+QI*WI)]
        SACH :PR:,1     PR = (1/2)[PR+(QR*WR+QI*WI)]
        SUBH :QR:       ACC= (1/4)[PR-(QR*WR+QI*WI)]
        SACH :QR:,1     QR = (1/2)[PR-(QR*WR+QI*WI)]
*
*       CALCULATE Im[Pm+1] & Im[Qm+1] STORE RESULTS in PI & QI
*
        LAC  :PI:,14    ACC= (1/4)PI
        ADD  :QI:,15    ACC= (1/4)[PI+(QI*WR-QR*WI)]
        SACH :PI:,1     PI = (1/2)[PI+(QI*WRRQR*WI)]
        SUBH :QI:       ACC= (1/4)[PI-(QI*WR-QR*WI)]
        SACH :QI:,1     QI = (1/2)[PI-(QI*WR-QR*WI)]
        $END


*************************************************************
*                                                           *
*       MACRO 9:                                            *
*                                                           *
*                     A     A'         B1                   *
*                      \ /   \     /                        *
*                      / \     \ /    B2                    *
*                     B     B'\ / \ /                       *
*                     C     C'/ \ / \                       *
*                      \ /     / \    B3                    *
*                      / \   /     \                        *
*                     D     D'         B4                   *
*                                                           *
*                                                           *
*       A = R1+jI1                                          *
*                                                           *
*       B = R2+jI2                                          *
*                                                           *
*       C = R3+jI3                                          *
*                                                           *
*       D = R4+jI4                                          *
*                                                           *
*       A' = (R1+R2) + j(I1+I2)                             *
*                                                           *
*       B' = (R1-R2) + j(I1-I2)                             *
*                                                           *
*       C' = (R3+R4) + j(R3+R4)                             *
*                                                           *
```

```
*       D' = (R3-R4) + j(R3-R4)                                *
*                                                              *
*       B1 = A'+C'                                             *
*                                                              *
*       B2 = B'-jD'                                            *
*                                                              *
*       B3 = A'-C'                                             *
*                                                              *
*       B4 = B'+jD'                                            *
*                                                              *
*       REAL(B1) = ((R1+R2) + (R3+R4)) /2                      *
*       IMAG(B1) = ((I1+I2) + (I3+I4)) /2                      *
*                                                              *
*       REAL(B2) = ((R1-R2) + (I3-I4)) /2                      *
*       IMAG(B2) = ((I1-I2) - (R3-R4)) /2                      *
*                                                              *
*       REAL(B3) = ((R1+R2) - (R3+R4)) /2                      *
*       IMAG(B3) = ((I1+I2) - (I3+I4)) /2                      *
*                                                              *
*       REAL(B4) = ((R1-R2) - (I3-I4)) /2                      *
*       IMAG(B4) = ((I1-I2) + (R3-R4)) /2                      *
*                                                              *
*                                                              *
* THE FIRST TWO STAGES OF A RADIX-2 N-POINT DIT FFT CAN BE     *
* IMPLEMENTED WITH A SPECIAL RADIX-4 BUTTERFLY WHICH HAS A     *
* UNITY TWIDDLE FACTOR USED TO SPEED UP THE EXECUTION TIME     *
* OF THE FFT WITH THE ABOVE EQUATIONS.  WHEN USING THESE       *
* EQUATIONS, ALL INPUT VALUES MUST BE WITHIN THE RANGE         *
* -1 <= X < 1.0. TOTAL NUMBER OF INTRUCTIONS IS 37.            *
* EXECUTION TIME IS EQUIVALENT TO 39 MACHINE CYCLES.           *
*                                                              *
****************************************************************


****************************************************************
*                                                              *
*  THE FOLLOWING STEPS ARE USED TO IMPLEMENT THE SPECIAL       *
*  RADIX-4 MACRO 'COMBO' FOR THE FIRST TWO STAGES OF AN        *
*  N-POINT RADIX-2 DIT FFT.                                    *
*                                                              *
*  STEP 1         STEP 2              STEP 3                   *
*  ------         ------              ------                   *
*                                                              *
*  R1 R1          (R1+R2)/1           [(R1+R2)+(R3+R4)]/2      *
*  I1 I1          (I1+I2)/1           [(I1+I2)+(I3+I4)]/2      *
*                                                              *
*  R2 R2          [(R1-R2)+(I3-I4)]/2 [(R1-R2)+(I3-I4)]/2      *
*  I2 I2          [(I1-I2)-(R3-R4)]/2 [(I1-I2)-(R3-R4)]/2      *
*                                                              *
*  R3 (R3+R4)/1   (R3+R4)/1           [(R1+R2)-(R3+R4)]/2      *
*  I3 (I3+I4)/1   (I3+I4)/1           [(I1+I2)-(I3+I4)]/2      *
*                                                              *
*  R4 (R3-R4)/1   [(R1-R2)-(I3-I4)]/2 [(R1-R2)-(I3-I4)]/2      *
*  I4 (I3-I4)/1   R4--->I4:=(R3-R4)/1                          *
*                 [(I1-I2)+(R3-R4)]/2 [(I1-I2)+(R3-R4)]/2      *
*                                                              *
****************************************************************
COMBO    $MACRO    R1,I1,R2,I2,R3,I3,R4,I4
*
*       CALCULATE PARTIAL TERMS FOR R3,R4,I3 AND I4
*
        LAC      :R3:,14   ACC := (1/4)(R3)
        ADD      :R4:,14   ACC := (1/4)(R3+R4)
        SACH     :R3:,1    R3  := (1/2)(R3+R4)
        SUB      :R4:,15   ACC := (1/4)(R3+R4)-(1/2)(R4)
        SACH     :R4:,1    R4  := (1/2)(R3-R4)
        LAC      :I3:,14   ACC := (1/4)(I3)
        ADD      :I4:,14   ACC := (1/4)(I3+I4)
        SACH     :I3:,1    I3  := (1/2)(I3+I4)
        SUB      :I4:,15   ACC := (1/4)(I3+I4)-(1/2)(I4)
        SACH     :I4:,1    I4  := (1/2)(I3-I4)
*
*       CALCULATE PARTIAL TERMS FOR R2,R4,I2 AND I4
*
        LAC      :R1:,14   ACC := (1/4)(R1)
        ADD      :R2:,14   ACC := (1/4)(R1+R2)
        SACH     :R1:,1    R1  := (1/2)(R1+R2)
        SUB      :R2:,15   ACC := (1/4)(R1+R2)-(1/2)(R2)
        ADD      :I4:,15   ACC := (1/4)[(R1-R2)+(I3-I4)]
        SACH     :R2:      R2  := (1/4)[(R1-R2)+(I3-I4)]
        SUBH     :I4:      ACC := (1/4)[(R1-R2)-(I3-I4)]
        DMOV     :R4:      I4  := R4 = (1/2)(R3-R4)
        SACH     :R4:      R4  := (1/4)[(R1-R2)-(I3-I4)]
        LAC      :I1:,14   ACC := (1/4)(I1)
        ADD      :I2:,14   ACC := (1/4)(I1+I2)
        SACH     :I1:,1    I1  := (1/2)(I1+I2)
        SUB      :I2:,15   ACC := (1/4)(I1+I2)-(1/2)(I2)
        SUB      :I4:,15   ACC := (1/4)[(I1-I2)-(R3-R4)]
        SACH     :I2:      I2  := (1/4)[(I1-I2)-(R3-R4)]
        ADDH     :I4:      ACC := (1/4)[(I1-I2)+(R3-R4)]
        SACH     :I4:      I4  := (1/4)[(I1-I2)+(R3-R4)]
*
*       CALCULATE PARTIAL TERMS FOR R1,R3,I1 AND I3
*

        LAC      :R1:,15   ACC := (1/4)(R1+R2).
        ADD      :R3:,15   ACC := (1/4)[(R1+R2)+(R3+R4)]
        SACH     :R1:      R1  := (1/4)[(R1+R2)+(R3+R4)]
        SUBH     :R3:      ACC := (1/4)[(R1+R2)-(R3+R4)]
        SACH     :R3:      R3  := (1/4)[(R1+R2)-(R3+R4)]
        LAC      :I1:,15   ACC := (1/4)(I1+I2)
        ADD      :I3:,15   ACC := (1/4)[(I1+I2)+(I3+I4)]
        SACH     :I1:      I1  := (1/4)[(I1+I2)+(I3+I4)]
        SUBH     :I3:      ACC := (1/4)[(I1+I2)-(I3+I4)]
        SACH     :I3:      I3  := (1/4)[(I1+I2)-(I3+I4)]
        $END
```

APPENDIX B
FFT MACRO LIBRARY (INDIRECT ADDRESSING)

APPENDIX B

```
*************************************************************
*                                                           *
* THIS TMS32020 FFT MACRO LIBRARY CONTAINS A TOTAL OF 7     *
* MACROS FOR IMPLEMENTING A GENERAL RADIX-2 DIT N-POINT     *
* FFT USING INDIRECT ADDRESS FORMAT OF THE TMS32020.  ALL   *
* DIT BUTTERFLIES ARE IMPLEMENTED WITH DYNAMIC SCALING TO   *
* AVOID ARITHMETIC OVERFLOWS.  2'S COMPLEMENT FIXED-POINT   *
* FRACTIONAL ARITHMETIC IS USED THROUGHOUT.  WHILE THESE    *
* MACROS ARE NOT NECESSARILY OPTIMISED FOR SPEED, THEY      *
* ARE SO STRUCTURED THAT NO INTERMEDIATE OR TEMPORARY       *
* REGISTERS ARE REQUIRED FOR THEIR EXECUTION.  HENCE,       *
* THESE MACROS ARE PARTICULARLY USEFUL IN APPLICATIONS      *
* WHERE MEMORY SPACE PROVES TO BE CRITICAL.  IN ADDITION,   *
* ALL FFT COMPUTATIONS ARE DONE IN PLACE AND THE REAL AND   *
* IMAGINARY PARTS OF ALL COMPLEX INPUTS ARE ASSUMED TO BE   *
* IN CONSECUTIVE DATA MEMORY LOCATIONS.                     *
*                                                           *
*************************************************************
*                                                           *
*************************************************************
*                                                           *
*                                        K                  *
*          P .                . P = P + Q*W                 *
*              `            '            N                  *
*                `        '                                 *
*                  `    '                                   *
*                   `  '                                    *
*                     O                                     *
*                   '  `                                    *
*                  '     `                                  *
*               ' K        `                                *
*             '  W           `           K                  *
*          Q .    N           . Q = P - Q*W                 *
*                                        N                  *
*                                                           *
*                                                           *
*        NOTATION FOR RADIX-2 DIT FFT BUTTERFLY             *
*                                                           *
*************************************************************
*                                                           *
*************************************************************
*                                                           *
*                                                         M *
* IN A RADIX-2 N-POINT FFT THERE ARE M PASSES WHERE N = 2   *
* WITH [N/2] 2-POINT BUTTERFLIES PER PASS, GIVING A TOTAL   *
* OF [N/2]LOG N BUTTERFLIES PER FFT.                        *
*            2                                              *
*                                                           *
*************************************************************
```

```
*        MACRO 1: W=1, k=0.                                 *
*                                                           *
*        P=(PR+jPI)      P+Q=(PR+QR)+j(PI+QI)               *
*                   \ /                                     *
*                   / \                                     *
*        Q=(QR+jQI)      P-Q=(PR-QR)+j(PI-QI)               *
*                                                           *
*           -j(2(pi)/N)k                                    *
*        W=e             =COS[(2(pi)/N)k]-jSIN[(2(pi)/N)k]  *
*                                                           *
*                     =WR+jWI =1                            *
*                                                           *
*  ALL OUTPUT SAMPLES ARE SCALED DOWN BY 2 TO ACCOMMODATE   *
*  A 1-BIT OVERFLOW.  HOWEVER, NO OVERFLOWS WILL OCCUR      *
*  FOR FRACTIONAL INPUTS OF THE FORM X : -1 <= X < 1.  A    *
*  TOTAL OF 12 INSTRUCTIONS ARE USED SUCH THAT EXECUTION    *
*  TIME IS EQUAL TO 14 MACHINE CYCLES.                      *
*                                                           *
*  MACRO ENTRY CONDITION: ARP MUST POINT AT AR1             *
*  MACRO EXIT CONDITION : ARP MUST POINT AT AR1             *
*                                                           *
*  MEMORY ADDRESS BIAS  : BIAS FOR DATA MEMORY PAGE 4       *
*                                                           *
*************************************************************
ZEROI    $MACRO   PR,QR,BIAS
*
*        INITIALISE AUXILIARY REGISTERS
*
         LRLK     AR1,:PR:+:BIAS:    AR1 POINTS TO PR
         LRLK     AR2,:QR:+:BIAS:    AR2 POINTS TO QR
*
*        CALCULATE Re[P+Q] AND Re[P-Q]
*
         LAC      *,15,AR2      ACC := (1/2)(PR)
         ADD      *,15,AR1      ACC := (1/2)(PR+QR)
         SACH     *+,0,AR2      PR  := (1/2)(PR+QR)
         SUBH     *             ACC := (1/2)(PR+QR)-(QR)
         SACH     *+,0,AR1      QR  := (1/2)(PR-QR)
*
*        CALCULATE Im[P+Q] AND Im[P-Q]
*
         LAC      *,15,AR2      ACC := (1/2)(PI)
         ADD      *,15,AR1      ACC := (1/2)(PI+QI)
         SACH     *-,0,AR2      PI  := (1/2)(PI+QI)
         SUBH     *             ACC := (1/2)(PI+QI)-(QI)
         SACH     *-,0,AR1      QI  := (1/2)(PI-QI)
         $END
```

```
************************************************************
*                                                          *
*       MACRO 2: W=-j, k=N/4.                              *
*                                                          *
*       P=(PR+jPI)      P-jQ=(PR+QI)+j(PI-QR)              *
*                  \ /                                     *
*                  / \                                     *
*       Q=(QR+jQI)      P+jQ=(PR-QI)+j(PI+QR)              *
*                                                          *
*          -j(2(pi)/N)k                                    *
*       W=e            =COS[(2(pi)/N)k]-jSIN[(2(pi)/N)k]   *
*                                                          *
*                      =WR+jWI =-j                         *
*                                                          *
*  ALL OUTPUT SAMPLES ARE SCALED DOWN BY 2 TO ACCOMMODATE  *
*  A 1-BIT OVERFLOW.  HOWEVER, NO OVERFLOWS WILL OCCUR     *
*  FOR FRACTIONAL INPUTS OF THE FORM X : -1 <= X < 1.  A   *
*  TOTAL OF 13 INSTRUCTIONS ARE USED SUCH THAT EXECUTION   *
*  TIME IS EQUAL TO 15 MACHINE CYCLES.                     *
*                                                          *
*  WARNING: THIS MACRO REQUIRES THE INPUT SAMPLES QR AND   *
*           QI TO BE IN CONSECUTIVE DATA MEMORY LOCATIONS  *
*           ASCENDING ORDER. THE FOLLOWING STEPS ARE USED  *
*           TO IMPLEMENT THIS MACRO:                       *
*                                                          *
*          (1) [PI-QR] -----> [PI]                         *
*          (2) [PI+QR] -----> [QR]                         *
*          (3) [PR+QI] -----> [PR]                         *
*          (4) [PR-QI] -----> [ACC]                        *
*          (5) [QR]    -----> [QI]                         *
*          (6) [ACC]   -----> [QR]                         *
*                                                          *
*  MACRO ENTRY CONDITION: ARP MUST POINT AT AR1            *
*  MACRO EXIT CONDITION : ARP MUST POINT AT AR1            *
*                                                          *
*  MEMORY ADDRESS BIAS   : BIAS FOR DATA MEMORY PAGE 4     *
*                                                          *
************************************************************
PBY2I   $MACRO   PR,QR,BIAS
*
*       INITIALISE AUXILIARY REGISTERS
*
        LRLK    AR1,:PR:+:BIAS:+1  AR1 POINTS TO PI
        LRLK    AR2,:QR:+:BIAS:    AR2 POINTS TO QR
*
*       CALCULATE Re[P+jQ] AND Re[P-jQ]
*
        LAC     *,15,AR2     ACC := (1/2)(PI)
        SUB     *,15,AR1     ACC := (1/2)(PI-QR)
        SACH    *-,0,AR2     PI  := (1/2)(PI-QR)
        ADDH    *            ACC := (1/2)(PI-QR)+(QR)
        SACH    *+,0,AR1     QR  := (1/2)(PI+QR)
*
*       CALCULATE Im[P+jQ] AND Im[P-jQ]
*
        LAC     *,15,AR2     ACC := (1/2)(PR)
        ADD     *,15,AR1     ACC := (1/2)(PR+QI)
        SACH    *,0,AR2      PR  := (1/2)(PR+QI)
```

```
        SUBH    *-           ACC := (1/2)(PR+QI)-(QI)
        DMOV    *            QR  -> QI
        SACH    *,0,AR1      QR  := (1/2)(PR-QI)
        $END

************************************************************
*                                                          *
*       MACRO 3:     k=N/8.                                *
*                                                          *
*       P=(PR+jPI)   P+Q*W=(PR+Re[Q*W])+j(PI+Im[Q*W])      *
*                 \ /                                      *
*                 / \                                      *
*       Q=(QR+jQI)   P-Q*W=(PR-Re[Q*W])+j(PI-Im[Q*W])      *
*                                                          *
*          -j(2(pi)/N)k                                    *
*     W =e              =COS((pi)/4)-jSIN((pi)/4)=WR+jWI   *
*      N                                                   *
*                     =WR+jWI                              *
*                                                          *
*       LET          W=|COS((pi)/4)|=|SIN((pi)/4)|         *
*                                                          *
*       THEN     [Q*W]=(QR+QI)*W+j(QI-QR)*W                *
*             Re[Q*W]=(QI+QR)*W                            *
*             Im[Q*W]=(QI-QR)*W                            *
*                                                          *
*  ALL OUTPUT SAMPLES ARE SCALED DOWN BY 2 TO ACCOMMODATE  *
*  A 1-BIT OVERFLOW.  HOWEVER, NO OVERFLOWS WILL OCCUR     *
*  FOR FRACTIONAL INPUTS OF THE FORM X : -1 <= X < 1.  A   *
*  TOTAL OF 22 INSTRUCTIONS ARE USED SUCH THAT EXECUTION   *
*  TIME IS EQUAL TO 24 MACHINE CYCLES. THIS MACRO REQUIRES *
*  W TO BE THE ABSOLUTE VALUE (MAGNITUDE) OF COS( /4) AND  *
*  SIN( /4).  THE SIGNS OF THESE TRIG FUNCTIONS HAVE BEEN  *
*  TAKEN CARE OF IN THE CODE.                              *
*                                                          *
*  MACRO ENTRY CONDITION: ARP MUST POINT AT AR1            *
*  MACRO EXIT CONDITION : ARP MUST POINT AT AR1            *
*                                                          *
*  MEMORY ADDRESS BIAS   : BIAS FOR DATA MEMORY PAGE 4     *
*                                                          *
************************************************************
PBY4I   $MACRO    PR,QR,BIAS
*
        LRLK     AR1,:QR:+:BIAS:+1  AR1 POINTS TO QI
        LRLK     AR2,:PR:+:BIAS:    AR2 POINTS TO PR
*
        LAC      *-,14        ACC := (1/4)(QI)
        SUB      *+,14        ACC := (1/4)(QI-QR)
        SACH     *-,1         QI  := (1/2)(QI-QR)
        ADD      *,15         ACC := (1/4)(QI+QR)
        SACH     *,1,AR2      QR  := (1/2)(QI+QR)
        LAC      *,14,AR0     ACC := (1/4)(PR)
        LT       *,AR1        T   :=W=COS(PI/4)=SIN(PI/4)
        MPY      *,AR2        P   := (1/4)(QI+QR)*W
        APAC                  ACC := (1/4)[PR+(QI+QR)*W]
        SACH     *+,1,AR1     PR  := (1/2)[PR+(QI+QR)*W]
        SPAC                  ACC := (1/4)(PR)
        SPAC                  ACC := (1/4)[PR-(QI+QR)*W]
        SACH     *+,1,AR2     QR  := (1/2)[PR-(QI+QR)*W]
        LAC      *,14,AR1     ACC := (1/4)(PI)
```

```
        MPY     *,AR2      P   := (1/4)(QI-QR)*W
        APAC               ACC := (1/4)[PI+(QI-QR)*W]
        SACH    *-,1,AR1   PI  := (1/2)[PI+(QI-QR)*W]
        SPAC               ACC := (1/4)(PI)
        SPAC               ACC := (1/4)[PI-(QI-QR)*W]
        SACH    *-,1,AR1   QI  := (1/2)[PI-(QI-QR)*W]
        $END


************************************************************
*                                                          *
*       MACRO 4:     k=3N/8.                               *
*                                                          *
*       P=(PR+jPI)   P+Q*W=(PR+Re[Q*W])+j(PI+Im[Q*W])      *
*                 \ /                                      *
*                 / \                                      *
*       Q=(QR+jQI)   P-Q*W=(PR-Re[Q*W])+j(PI+Im[Q*W])      *
*                                                          *
*           -j(2(pi)/N)k                                   *
*       W =e            =COS(3(pi)/4)-jSIN(3(pi)/4)        *
*        N                                                 *
*                     =WR+jWI                              *
*                                                          *
*       LET          W=|COS(3(pi)/4)|=|SIN(3(pi)/4)|       *
*                                                          *
*       THEN     [Q*W]=(QI-QR)*W-j(QI+QR)*W                *
*              RE[Q*W]=(QI-QR)*W                           *
*              IM[Q*W]=(QI+QR)*W                           *
*                                                          *
*  ALL OUTPUT SAMPLES ARE SCALED DOWN BY 2 TO ACCOMMODATE  *
*  A 1-BIT OVERFLOW.  HOWEVER, NO OVERFLOWS WILL OCCUR     *
*  FOR FRACTIONAL INPUTS OF THE FORM X : -1 <= X < 1.  A   *
*  TOTAL OF 22 INSTRUCTIONS ARE USED SUCH THAT EXECUTION   *
*  TIME IS EQUAL TO 24 MACHINE CYCLES. THIS MACRO REQUIRES *
*  W TO BE THE ABSOLUTE VALUE (MAGNITUDE) OF COS(3 /4) AND *
*  SIN(3 /4).  THE SIGNS OF THESE TRIG FUNCTIONS HAVE BEEN *
*  TAKEN CARE OF IN THE CODE.                              *
*                                                          *
*  MACRO ENTRY CONDITION: ARP MUST POINT AT AR1            *
*  MACRO EXIT CONDITION : ARP MUST POINT AT AR1            *
*                                                          *
*  MEMORY ADDRESS BIAS   : BIAS FOR DATA MEMORY PAGE 4     *
*                                                          *
************************************************************
P3BY4I  $MACRO   PR,QR,BIAS
*
        LRLK     AR1,:QR:+:BIAS:+1  AR1 POINTS TO QI
        LRLK     AR2,:PR:+:BIAS:    AR2 POINTS TO PR
*
        LAC     *-,14      ACC := (1/4)(QI)
        SUB     *+,14      ACC := (1/4)(QI-QR)
        SACH    *-,1       QI  := (1/2)(QI-QR)
        ADD     *,15       ACC := (1/4)(QI+QR)
        SACH    *+,1,AR2   QR  := (1/2)(QI+QR)
        LAC     *,14,AR0   ACC := (1/4)(PR)
        LT      *,AR1      T   :=W=COS( /4)=SIN( /4)
        MPY     *-,AR2     P   := (1/4)(QI-QR)*W
        APAC               ACC := (1/4)[PR+(QI-QR)*W]
        SACH    *+,1,AR1   PR  := (1/2)[PR+(QI-QR)*W]
```

```
        SPAC               ACC := (1/4)(PR)
        SPAC               ACC := (1/4)[PR-(QI-QR)*W]
        MPY     *          P   := (1/4)(QI+QR)*W
        SACH    *+,1,AR2   QR  := (1/2)[PR-(QI-QR)*W]
        LAC     *,14       ACC := (1/4)(PI)
        SPAC               ACC := (1/4)[PI-(QI+QR)*W]
        SACH    *-,1,AR1   PI  := (1/2)[PI-(QI+QR)*W]
        APAC               ACC := (1/4)(PI)
        APAC               ACC := (1/4)[PI+(QI+QR)*W]
        SACH    *-,1,AR1   QI  := (1/2)[PI+(QI+QR)*W]
        $END


************************************************************
*                                                          *
*       MACRO 5:                                           *
*                                                          *
*                    A     A'       B1                     *
*                     \ /   \     /                        *
*                     / \     \ /   B2                     *
*                    B     B'\ / \ /                       *
*                    C     C'/ \ / \                       *
*                     \ /     / \   B3                     *
*                     / \   /     \                        *
*                    D     D'       B4                     *
*                                                          *
*       A = R1+jI1                                         *
*                                                          *
*       B = R2+jI2                                         *
*                                                          *
*       C = R3+jI3                                         *
*                                                          *
*       D = R4+jI4                                         *
*                                                          *
*       A' = (R1+R2) + j(I1+I2)                            *
*                                                          *
*       B' = (R1-R2) + j(I1-I2)                            *
*                                                          *
*       C' = (R3+R4) + j(R3+R4)                            *
*                                                          *
*       D' = (R3-R4) + j(R3-R4)                            *
*                                                          *
*       B1 = A'+C'                                         *
*                                                          *
*       B2 = B'-jD'                                        *
*                                                          *
*       B3 = A'-C'                                         *
*                                                          *
*       B4 = B'+jD'                                        *
*                                                          *
*       REAL(B1) = ((R1+R2) + (R3+R4)) /2                  *
*       IMAG(B1) = ((I1+I2) + (I3+I4)) /2                  *
*                                                          *
*       REAL(B2) = ((R1-R2) + (I3-I4)) /2                  *
*       IMAG(B2) = ((I1-I2) - (R3-R4)) /2                  *
*                                                          *
*       REAL(B3) = ((R1+R2) - (R3+R4)) /2                  *
*       IMAG(B3) = ((I1+I2) - (I3+I4)) /2                  *
*                                                          *
```

```
*       REAL(B4) = ((R1-R2) - (I3-I4)) /2                       *
*       IMAG(B4) = ((I1-I2) + (R3-R4)) /2                       *
*                                                               *
*                                                               *
* THE FIRST TWO STAGES OF A RADIX-2 N-POINT DIT FFT CAN BE *
* IMPLEMENTED WITH A SPECIAL RADIX-4 BUTTERFLY WHICH HAS A *
* UNITY TWIDDLE FACTOR USED TO SPEED UP THE EXECUTION TIME *
* OF THE FFT WITH THE ABOVE EQUATIONS. WHEN USING THESE    *
* EQUATIONS, ALL INPUT VALUES MUST BE WITHIN THE RANGE     *
* -1 <= X < 1.0.  TOTAL NUMBER OF INTRUCTIONS IS 39.       *
* EXECUTION TIME IS EQUIVALENT TO 41 MACHINE CYCLES.       *
*                                                               *
*****************************************************************


*****************************************************************
*  THE FOLLOWING STEPS ARE USED TO IMPLEMENT THE SPECIAL   *
*  RADIX-4 MACRO 'COMBO' FOR THE FIRST TWO STAGES OF AN    *
*  N-POINT RADIX-2 DIT FFT.                                *
*                                                          *
*  STEP 1        STEP 2              STEP 3                *
*  ------        ------              ------                *
*                                                          *
*  R1 R1         (R1+R2)/1           [(R1+R2)+(R3+R4)]/2  *
*  I1 I1         (I1+I2)/1           [(I1+I2)+(I3+I4)]/2  *
*                                                          *
*  R2 R2         [(R1-R2)+(I3-I4)]/2  [(R1-R2)+(I3-I4)]/2  *
*  I2 I2         [(I1-I2)-(R3-R4)]/2  [(I1-I2)-(R3-R4)]/2  *
*                                                          *
*  R3 (R3+R4)/1  (R3+R4)/1           [(R1+R2)-(R3+R4)]/2  *
*  I3 (I3+I4)/1  (I3+I4)/1           [(I1+I2)-(I3+I4)]/2  *
*                                                          *
*  R4 (R3-R4)/1  [(R1-R2)-(I3-I4)]/2  [(R1-R2)-(I3-I4)]/2  *
*  I4 (I3-I4)/1  R4--->I4:=(R3-R4)/1                       *
*                [(I1-I2)+(R3-R4)]/2  [(I1-I2)+(R3-R4)]/2  *
*                                                          *
*   IN THE FOLLOWING CODE, ALL IMAGINARY TERMS (I'S) ARE   *
*   ASSUMED TO IN LOCATIONS CONSECUTIVE TO THOSE OF THE    *
*   REAL TERMS (R'S).  THEREFORE, THE ADDRESS OF EACH      *
*   IMAGINARY TERM (I) IS REPRESENTED AS (R+1).            *
*                                                          *
*****************************************************************
COMBOI  $MACRO    R1,R2,R3,R4,BIAS
*
*       CALCULATE PARTIAL TERMS FOR R3,R4,I3 AND I4
*
        LAC     :R3:-:BIAS:,14    ACC := (1/4)(R3)
        ADD     :R4:-:BIAS:,14    ACC := (1/4)(R3+R4)
        SACH    :R3:-:BIAS:,1     R3  := (1/2)(R3+R4)
        SUB     :R4:-:BIAS:,15    ACC := (1/4)(R3+R4)-(1/2)(R4)
        SACH    :R4:-:BIAS:,1     R4  := (1/2)(R3-R4)
        LAC     :R3:+1-:BIAS:,14  ACC := (1/4)(I3)
        ADD     :R4:+1-:BIAS:,14  ACC := (1/4)(I3+I4)
        SACH    :R3:+1-:BIAS:,1   I3  := (1/2)(I3+I4)
        SUB     :R4:+1-:BIAS:,15  ACC := (1/4)(I3+I4)-(1/2)(I4)
        SACH    :R4:+1-:BIAS:,1   I4  := (1/2)(I3-I4)
*
*       CALCULATE PARTIAL TERMS FOR R2,R4,I2 AND I4
*
        LAC     :R1:-:BIAS:,14     ACC := (1/4)(R1)
        ADD     :R2:-:BIAS:,14     ACC := (1/4)(R1+R2)
        SACH    :R1:-:BIAS:,1      R1  := (1/2)(R1+R2)
        SUB     :R2:-:BIAS:,15     ACC := (1/4)(R1+R2)-(1/2)(R2)
        ADD     :R4:+1-:BIAS:,15   ACC := (1/4)[(R1-R2)+(I3-I4)]
        SACH    :R2:-:BIAS:        R2  := (1/4)[(R1-R2)+(I3-I4)]
        SUBH    :R4:+1-:BIAS:      ACC := (1/4)[(R1-R2)-(I3-I4)]
        DMOV    :R4:-:BIAS:        I4  := R4 = (1/2)(R3-R4)
        SACH    :R4:-:BIAS:        R4  := (1/4)[(R1-R2)-(I3-I4)]
        LAC     :R1:+1-:BIAS:,14   ACC := (1/4)(I1)
        ADD     :R2:+1-:BIAS:,14   ACC := (1/4)(I1+I2)
        SACH    :R1:+1-:BIAS:,1    I1  := (1/2)(I1+I2)
        SUB     :R2:+1-:BIAS:,15   ACC := (1/4)(I1+I2)-(1/2)(I2)
        SUB     :R4:+1-:BIAS:,15   ACC := (1/4)[(I1-I2)-(R3-R4)]
        SACH    :R2:+1-:BIAS:      I2  := (1/4)[(I1-I2)-(R3-R4)]
        ADDH    :R4:+1-:BIAS:      ACC := (1/4)[(I1-I2)+(R3-R4)]
        SACH    :R4:+1-:BIAS:      I4  := (1/4)[(I1-I2)+(R3-R4)]

*
*       CALCULATE PARTIAL TERMS FOR R1,R3,I1 AND I3
*
        LAC     :R1:-:BIAS:,15     ACC := (1/4)(R1+R2).
        ADD     :R3:-:BIAS:,15     ACC := (1/4)[(R1+R2)+(R3+R4)]
        SACH    :R1:-:BIAS:        R1  := (1/4)[(R1+R2)+(R3+R4)]
        SUBH    :R3:-:BIAS:        ACC := (1/4)[(R1+R2)-(R3+R4)]
        SACH    :R3:-:BIAS:        R3  := (1/4)[(R1+R2)-(R3+R4)]
        LAC     :R1:+1-:BIAS:,15   ACC := (1/4)(I1+I2)
        ADD     :R3:+1-:BIAS:,15   ACC := (1/4)[(I1+I2)+(I3+I4)]
        SACH    :R1:+1-:BIAS:      I1  := (1/4)[(I1+I2)+(I3+I4)]
        SUBH    :R3:+1-:BIAS:      ACC := (1/4)[(I1+I2)-(I3+I4)]
        SACH    :R3:+1-:BIAS:      I3  := (1/4)[(I1+I2)-(I3+I4)]
        $END

*****************************************************************
*                                                               *
*  MACRO 6:  A GENERAL RADIX-2 DIT FFT 'BUTTERFLY'              *
*                                                               *
*       P=(PR+jQI)   P+Q*W=(PR+Re[Q*W])+j(PI+Im[Q*W])           *
*                  \ /                                          *
*                  / \                                          *
*       Q=(QR+jQI)   P-Q*W=(PR-Re[Q*W])+j(PI+Im[Q*W])           *
*                                                               *
*            -j(2(pi)/N)k                                       *
*       W =e              =COS(O)-jSIN(O)                       *
*        N                                                      *
*                         =WR+jWI                               *
*                                                               *
*  ALL OUTPUT SAMPLES ARE SCALED DOWN BY 2 TO ACCOMMODATE       *
*  A 1-BIT OVERFLOW.  HOWEVER, NO OVERFLOWS WILL OCCUR          *
*  FOR FRACTIONAL INPUTS OF THE FORM X : -1 <= X < 1.  A        *
*  TOTAL OF 25 INSTRUCTIONS ARE USED SUCH  THAT EXECUTION       *
*  TIME IS EQUAL TO 27 MACHINE CYCLES.   THIS MACRO             *
*  REQUIRES W TO BE THE ABSOLUTE VALUE (MAGNITUDE)  OF          *
*  COS(O) AND SIN(O).  THE SIGNS OF THESE TRIG FUNCTIONS        *
*  HAVE BEEN TAKEN CARE OF IN THE CODE.                         *
```

```
*                                                            *
*  MACRO ENTRY CONDITION: ARP MUST POINT AT AR1              *
*  MACRO EXIT CONDITION : ARP MUST POINT AT AR1              *
*                                                            *
*  MEMORY ADDRESS BIAS   : BIAS FOR DATA MEMORY PAGE 4       *
*                                                            *
**************************************************************
BTRFLI  $MACRO    PR,QR,WR,WI,BIAS
*
*       INITIALISE AUXILIARY REGISTERS
*
        LRLK      AR1,:QR:+:BIAS:+1  AR1 POINTS TO QI
        LRLK      AR2,:PR:+:BIAS:    AR2 POINTS TO PR
*
*       CALCULATE QR*WR + QI*WI AND STORE RESULT IN QI
*
        LT        *-          T-REGISTER := QI
        MPYK      :WI:        P-REGISTER := (1/16)(QI*WI)
        LTP       *           ACC := (1/16)(QI*WI); T-REGISTER=QR
        MPYK      :WR:        P-REGISTER := (1/16)(QR*WR)
        APAC                  ACC := (1/16)(QR*WR+QI*WI)
        SFR                   ACC := (1/32)(QR*WR+QI*WI)
        SACH      *+,4        QR  := (1/2)(QR*WR+QI*WI)


*
*       CALCULATE QI*WR - QR*WI AND STORE RESULT IN QR
*
        MPYK      -:WI:       P-REGISTER := (1/16)(-QR*WI)
        LTP       *           ACC := (1/16)(-QR*WI); T-REGISTER=QI
        MPYK      :WR:        P-REGISTER := (1/16)(QI*WR)
        APAC                  ACC := (1/16)(QI*WR-QR*WI)
        SFR                   ACC := (1/32)(QI*WR-QR*WI)
        SACH      *-,4,AR2    QI  := (1/2)(QI*WR-QR*WI)
*
*       CALCULATE Re[P+jQ] & Re[P-jQ] STORE RESULTS in PR & QR
*
        LAC       *,14,AR1    ACC := (1/4)PR
        ADD       *,15,AR2    ACC := (1/4)[PR+(QR*WR+QI*WI)]
        SACH      *+,1,AR1    PR  := (1/2)[PR+(QR*WR+QI*WI)]
        SUBH      *           ACC := (1/4)[PR-(QR*WR+QI*WI)]
        SACH      *+,1,AR2    QR  := (1/2)[PR-(QR*WR+QI*WI)]
*
*       CALCULATE Im[P+jQ] & Im[P-jQ] STORE RESULTS in PI & QI
*
        LAC       *,14,AR1    ACC := (1/4)PI
        ADD       *,15,AR2    ACC := (1/4)[PI+(QI*WR-QR*WI)]
        SACH      *-,1,AR1    PI  := (1/2)[PI+(QI*WR-QR*WI)]
        SUBH      *           ACC := (1/4)[PI-(QI*WR-QR*WI)]
        SACH      *-,1,AR1    QI  := (1/2)[PI-(QI*WR-QR*WI)]
        $END

**************************************************************
*                                                            *
*       MACRO 7:  RADIX-2 INPUT BIT REVERSAL                 *
*                                                            *
*       ENTRY CONDITION: ARP MUST POINT AT AR1               *
*                                                            *
*       EXIT  CONDITION: ARP MUST POINT AT AR1               *
*                                                            *
*       A TOTAL OF 10 INSTRUCTIONS ARE USED SUCH THAT        *
*       EXECUTION TIME IS EQUAL TO 12 MACHINE CYCLES.        *
*                                                            *
**************************************************************
*
BITRVI  $MACRO    PR,QR,BIAS
*
*       INITIALISE AUXILIARY REGISTERS
*
        LRLK      AR1,:PR:+:BIAS:  AR1 POINTS TO PR
        LRLK      AR2,:QR:+:BIAS:  AR2 POINTS TO QR
*
        ZALH      *,AR2
        ADDS      *,AR1
        SACL      *+,0,AR2
        SACH      *+,0,AR1
        ZALH      *,AR2
        ADDS      *,AR1
        SACL      *-,0,AR2
        SACH      *-,0,AR1
        $END
```

APPENDIX C
A 256-POINT, RADIX-2 DIT FFT IMPLEMENTATION

```
        A P P E N D I X   C
        -------------------

        IDT       'FFT256'
*************************************************************
*                                                           *
*   A 256-POINT RADIX-2 DIT COMPLEX FFT FOR THE TMS32020    *
*   ---------------------------------------------------     *
*                                                           *
*   THE FOLLOWING FILE RAD2FFT.MAC CONSISTS OF ALL THE      *
*   MACROS LISTED IN APPENDIX  B                            *
*                                                           *
*************************************************************
*
        COPY      RAD2FFT.MAC
*
*************************************************************
*                                                           *
*   DATA MEMORY MAP FOR PAGES 4, 5, 6 AND 7 (BLOCKS B0,B1)  *
*                                                           *
*************************************************************
*
        DORG     0
*
*       DATA MEMORY PAGE 4 (STARTING ADDRESS 512 OR >200)
*
X000    DATA     0,0
X001    DATA     0,0
X002    DATA     0,0
X003    DATA     0,0
X004    DATA     0,0
X005    DATA     0,0
X006    DATA     0,0
X007    DATA     0,0
X008    DATA     0,0
X009    DATA     0,0
X010    DATA     0,0
X011    DATA     0,0
X012    DATA     0,0
X013    DATA     0,0
X014    DATA     0,0
X015    DATA     0,0
X016    DATA     0,0
X017    DATA     0,0
X018    DATA     0,0
X019    DATA     0,0
X020    DATA     0,0
X021    DATA     0,0
X022    DATA     0,0
X023    DATA     0,0
X024    DATA     0,0
X025    DATA     0,0
X026    DATA     0,0
X027    DATA     0,0
X028    DATA     0,0
X029    DATA     0,0
X030    DATA     0,0
X031    DATA     0,0
X032    DATA     0,0
X033    DATA     0,0
X034    DATA     0,0
X035    DATA     0,0
X036    DATA     0,0
X037    DATA     0,0
X038    DATA     0,0
X039    DATA     0,0
X040    DATA     0,0
X041    DATA     0,0
X042    DATA     0,0
X043    DATA     0,0
X044    DATA     0,0
X045    DATA     0,0
X046    DATA     0,0
X047    DATA     0,0
X048    DATA     0,0
X049    DATA     0,0
X050    DATA     0,0
X051    DATA     0,0
X052    DATA     0,0
X053    DATA     0,0
X054    DATA     0,0
X055    DATA     0,0
X056    DATA     0,0
X057    DATA     0,0
X058    DATA     0,0
X059    DATA     0,0
X060    DATA     0,0
X061    DATA     0,0
X062    DATA     0,0
X063    DATA     0,0
*
*       DATA MEMORY PAGE 5 (STARTING ADDRESS 640 OR >280)
*
X064    DATA     0,0
X065    DATA     0,0
X066    DATA     0,0
X067    DATA     0,0
X068    DATA     0,0
X069    DATA     0,0
X070    DATA     0,0
X071    DATA     0,0
X072    DATA     0,0
X073    DATA     0,0
X074    DATA     0,0
X075    DATA     0,0
X076    DATA     0,0
X077    DATA     0,0
X078    DATA     0,0
X079    DATA     0,0
X080    DATA     0,0
X081    DATA     0,0
X082    DATA     0,0
X083    DATA     0,0
X084    DATA     0,0
X085    DATA     0,0
X086    DATA     0,0
X087    DATA     0,0
X088    DATA     0,0
X089    DATA     0,0
X090    DATA     0,0
```

```
X091      DATA    0,0
X092      DATA    0,0
X093      DATA    0,0
X094      DATA    0,0
X095      DATA    0,0
X096      DATA    0,0
X097      DATA    0,0
X098      DATA    0,0
X099      DATA    0,0
X100      DATA    0,0
X101      DATA    0,0
X102      DATA    0,0
X103      DATA    0,0
X104      DATA    0,0
X105      DATA    0,0
X106      DATA    0,0
X107      DATA    0,0
X108      DATA    0,0
X109      DATA    0,0
X110      DATA    0,0
X111      DATA    0,0
X112      DATA    0,0
X113      DATA    0,0
X114      DATA    0,0
X115      DATA    0,0
X116      DATA    0,0
X117      DATA    0,0
X118      DATA    0,0
X119      DATA    0,0
X120      DATA    0,0
X121      DATA    0,0
X122      DATA    0,0
X123      DATA    0,0
X124      DATA    0,0
X125      DATA    0,0
X126      DATA    0,0
X127      DATA    0,0
*
*         DATA MEMORY PAGE 6 (STARTING ADDRESS 768 OR >300)
*
X128      DATA    0,0
X129      DATA    0,0
X130      DATA    0,0
X131      DATA    0,0
X132      DATA    0,0
X133      DATA    0,0
X134      DATA    0,0
X135      DATA    0,0
X136      DATA    0,0
X137      DATA    0,0
X138      DATA    0,0
X139      DATA    0,0
X140      DATA    0,0
X141      DATA    0,0
X142      DATA    0,0
X143      DATA    0,0
X144      DATA    0,0
X145      DATA    0,0
X146      DATA    0,0
X147      DATA    0,0
X148      DATA    0,0
X149      DATA    0,0
X150      DATA    0,0
X151      DATA    0,0
X152      DATA    0,0
X153      DATA    0,0
X154      DATA    0,0
X155      DATA    0,0
X156      DATA    0,0
X157      DATA    0,0
X158      DATA    0,0
X159      DATA    0,0
X160      DATA    0,0
X161      DATA    0,0
X162      DATA    0,0
X163      DATA    0,0
X164      DATA    0,0
X165      DATA    0,0
X166      DATA    0,0
X167      DATA    0,0
X168      DATA    0,0
X169      DATA    0,0
X170      DATA    0,0
X171      DATA    0,0
X172      DATA    0,0
X173      DATA    0,0
X174      DATA    0,0
X175      DATA    0,0
X176      DATA    0,0
X177      DATA    0,0
X178      DATA    0,0
X179      DATA    0,0
X180      DATA    0,0
X181      DATA    0,0
X182      DATA    0,0
X183      DATA    0,0
X184      DATA    0,0
X185      DATA    0,0
X186      DATA    0,0
X187      DATA    0,0
X188      DATA    0,0
X189      DATA    0,0
X190      DATA    0,0
X191      DATA    0,0
*
*         DATA MEMORY PAGE 7 (STARTING ADDRESS 896 OR >380)
*
X192      DATA    0,0
X193      DATA    0,0
X194      DATA    0,0
X195      DATA    0,0
X196      DATA    0,0
X197      DATA    0,0
X198      DATA    0,0
X199      DATA    0,0
X200      DATA    0,0
X201      DATA    0,0
X202      DATA    0,0
X203      DATA    0,0
X204      DATA    0,0
```

```
X205      DATA     0,0
X206      DATA     0,0
X207      DATA     0,0
X208      DATA     0,0
X209      DATA     0,0
X210      DATA     0,0
X211      DATA     0,0
X212      DATA     0,0
X213      DATA     0,0
X214      DATA     0,0
X215      DATA     0,0
X216      DATA     0,0
X217      DATA     0,0
X218      DATA     0,0
X219      DATA     0,0
X220      DATA     0,0
X221      DATA     0,0
X222      DATA     0,0
X223      DATA     0,0
X224      DATA     0,0
X225      DATA     0,0
X226      DATA     0,0
X227      DATA     0,0
X228      DATA     0,0
X229      DATA     0,0
X230      DATA     0,0
X231      DATA     0,0
X232      DATA     0,0
X233      DATA     0,0
X234      DATA     0,0
X235      DATA     0,0
X236      DATA     0,0
X237      DATA     0,0
X238      DATA     0,0
X239      DATA     0,0
X240      DATA     0,0
X241      DATA     0,0
X242      DATA     0,0
X243      DATA     0,0
X244      DATA     0,0
X245      DATA     0,0
X246      DATA     0,0
X247      DATA     0,0
X248      DATA     0,0
X249      DATA     0,0
X250      DATA     0,0
X251      DATA     0,0
X252      DATA     0,0
X253      DATA     0,0
X254      DATA     0,0
X255      DATA     0,0
*
**************************************************************
*                                                            *
*    DATA LOCATION IN BLOCK B2 FOR W=COS(45) OR SIN(45)      *
*                                                            *
**************************************************************
*
          DORG     96
W         DATA     0
```

```
*
**************************************************************
*                                                            *
*   13-BIT TWIDDLE FACTORS FOR 256-POINT COMPLEX FFT         *
*                                                            *
**************************************************************
*
C000      EQU      4095
C001      EQU      4094
C002      EQU      4091
C003      EQU      4085
C004      EQU      4076
C005      EQU      4065
C006      EQU      4052
C007      EQU      4036
C008      EQU      4017
C009      EQU      3996
C010      EQU      3973
C011      EQU      3948
C012      EQU      3920
C013      EQU      3889
C014      EQU      3857
C015      EQU      3822
C016      EQU      3784
C017      EQU      3745
C018      EQU      3703
C019      EQU      3659
C020      EQU      3612
C021      EQU      3564
C022      EQU      3513
C023      EQU      3461
C024      EQU      3406
C025      EQU      3349
C026      EQU      3290
C027      EQU      3229
C028      EQU      3166
C029      EQU      3102
C030      EQU      3035
C031      EQU      2967
C032      EQU      2896
C033      EQU      2824
C034      EQU      2751
C035      EQU      2675
C036      EQU      2598
C037      EQU      2520
C038      EQU      2440
C039      EQU      2359
C040      EQU      2276
C041      EQU      2191
C042      EQU      2106
C043      EQU      2019
C044      EQU      1931
C045      EQU      1842
C046      EQU      1751
C047      EQU      1660
C048      EQU      1567
C049      EQU      1474
C050      EQU      1380
C051      EQU      1285
C052      EQU      1189
```

```
C053     EQU     1092
C054     EQU     995
C055     EQU     897
C056     EQU     799
C057     EQU     700
C058     EQU     601
C059     EQU     501
C060     EQU     401
C061     EQU     301
C062     EQU     201
C063     EQU     101
C064     EQU     0
C065     EQU     -101
C066     EQU     -201
C067     EQU     -301
C068     EQU     -401
C069     EQU     -501
C070     EQU     -601
C071     EQU     -700
C072     EQU     -799
C073     EQU     -897
C074     EQU     -995
C075     EQU     -1092
C076     EQU     -1189
C077     EQU     -1285
C078     EQU     -1380
C079     EQU     -1474
C080     EQU     -1567
C081     EQU     -1660
C082     EQU     -1751
C083     EQU     -1842
C084     EQU     -1931
C085     EQU     -2019
C086     EQU     -2106
C087     EQU     -2191
C088     EQU     -2276
C089     EQU     -2359
C090     EQU     -2440
C091     EQU     -2520
C092     EQU     -2598
C093     EQU     -2675
C094     EQU     -2751
C095     EQU     -2824
C096     EQU     -2896
C097     EQU     -2967
C098     EQU     -3035
C099     EQU     -3102
C100     EQU     -3166
C101     EQU     -3229
C102     EQU     -3290
C103     EQU     -3349
C104     EQU     -3406
C105     EQU     -3461
C106     EQU     -3513
C107     EQU     -3564
C108     EQU     -3612
C109     EQU     -3659
C110     EQU     -3703
C111     EQU     -3745
C112     EQU     -3784
C113     EQU     -3822
C114     EQU     -3857
C115     EQU     -3889
C116     EQU     -3920
C117     EQU     -3948
C118     EQU     -3973
C119     EQU     -3996
C120     EQU     -4017
C121     EQU     -4036
C122     EQU     -4052
C123     EQU     -4065
C124     EQU     -4076
C125     EQU     -4085
C126     EQU     -4091
C127     EQU     -4094
*
S000     EQU     0
S001     EQU     101
S002     EQU     201
S003     EQU     301
S004     EQU     401
S005     EQU     501
S006     EQU     601
S007     EQU     700
S008     EQU     799
S009     EQU     897
S010     EQU     995
S011     EQU     1092
S012     EQU     1189
S013     EQU     1285
S014     EQU     1380
S015     EQU     1474
S016     EQU     1567
S017     EQU     1660
S018     EQU     1751
S019     EQU     1842
S020     EQU     1931
S021     EQU     2019
S022     EQU     2106
S023     EQU     2191
S024     EQU     2276
S025     EQU     2359
S026     EQU     2440
S027     EQU     2520
S028     EQU     2598
S029     EQU     2675
S030     EQU     2751
S031     EQU     2824
S032     EQU     2896
S033     EQU     2967
S034     EQU     3035
S035     EQU     3102
S036     EQU     3166
S037     EQU     3229
S038     EQU     3290
S039     EQU     3349
S040     EQU     3406
S041     EQU     3461
S042     EQU     3513
S043     EQU     3564
```

```
S044    EQU    3612
S045    EQU    3659
S046    EQU    3703
S047    EQU    3745
S048    EQU    3784
S049    EQU    3822
S050    EQU    3857
S051    EQU    3889
S052    EQU    3920
S053    EQU    3948
S054    EQU    3973
S055    EQU    3996
S056    EQU    4017
S057    EQU    4036
S058    EQU    4052
S059    EQU    4065
S060    EQU    4076
S061    EQU    4085
S062    EQU    4091
S063    EQU    4094
S064    EQU    4095
S065    EQU    4094
S066    EQU    4091
S067    EQU    4085
S068    EQU    4076
S069    EQU    4065
S070    EQU    4052
S071    EQU    4036
S072    EQU    4017
S073    EQU    3996
S074    EQU    3973
S075    EQU    3948
S076    EQU    3920
S077    EQU    3889
S078    EQU    3857
S079    EQU    3822
S080    EQU    3784
S081    EQU    3745
S082    EQU    3703
S083    EQU    3659
S084    EQU    3612
S085    EQU    3564
S086    EQU    3513
S087    EQU    3461
S088    EQU    3406
S089    EQU    3349
S090    EQU    3290
S091    EQU    3229
S092    EQU    3166
S093    EQU    3102
S094    EQU    3035
S095    EQU    2967
S096    EQU    2896
S097    EQU    2824
S098    EQU    2751
S099    EQU    2675
S100    EQU    2598
S101    EQU    2520
S102    EQU    2440
S103    EQU    2359
S104    EQU    2276
S105    EQU    2191
S106    EQU    2106
S107    EQU    2019
S108    EQU    1931
S109    EQU    1842
S110    EQU    1751
S111    EQU    1660
S112    EQU    1567
S113    EQU    1474
S114    EQU    1380
S115    EQU    1285
S116    EQU    1189
S117    EQU    1092
S118    EQU    995
S119    EQU    897
S120    EQU    799
S121    EQU    700
S122    EQU    601
S123    EQU    501
S124    EQU    401
S125    EQU    301
S126    EQU    201
S127    EQU    101
*
        AORG   0
        B      INIT
*
*       SYSTEM INITIALIZATION
*
        AORG   >20
*
*       16-BIT TWIDDLE FACTOR FOR SPECIAL MACROS
*
WVAL    DATA   >5A82
*
INIT    SPM    0
        CNFD
        ROVM
        SSXM
        LARP   AR0
        LRLK   AR0,W
        LALK   WVAL
        TBLR   *,AR1
*
*       FFT CODE WITH BIT-REVERSED INPUT SAMPLES
*
FFT256  BITRVI X001,X128,512
        BITRVI X002,X064,512
        BITRVI X003,X192,512
        BITRVI X004,X032,512
        BITRVI X005,X160,512
        BITRVI X006,X096,512
        BITRVI X007,X224,512
        BITRVI X008,X016,512
        BITRVI X009,X144,512
        BITRVI X010,X080,512
        BITRVI X011,X208,512
        BITRVI X012,X048,512
        BITRVI X013,X176,512
```

```
BITRVI    X014,X112,512
BITRVI    X015,X240,512
BITRVI    X017,X136,512
BITRVI    X018,X072,512
BITRVI    X019,X200,512
BITRVI    X020,X040,512
BITRVI    X021,X168,512
BITRVI    X022,X104,512
BITRVI    X023,X232,512
BITRVI    X025,X152,512
BITRVI    X026,X088,512
BITRVI    X027,X216,512
BITRVI    X028,X056,512
BITRVI    X029,X184,512
BITRVI    X030,X120,512
BITRVI    X031,X248,512
BITRVI    X033,X132,512
BITRVI    X034,X068,512
BITRVI    X035,X196,512
BITRVI    X037,X164,512
BITRVI    X038,X100,512
BITRVI    X039,X228,512
BITRVI    X041,X148,512
BITRVI    X042,X084,512
BITRVI    X043,X212,512
BITRVI    X044,X052,512
BITRVI    X045,X180,512
BITRVI    X046,X116,512
BITRVI    X047,X244,512
BITRVI    X049,X140,512
BITRVI    X050,X076,512
BITRVI    X051,X204,512
BITRVI    X053,X172,512
BITRVI    X054,X108,512
BITRVI    X055,X236,512
BITRVI    X057,X156,512
BITRVI    X058,X092,512
BITRVI    X059,X220,512
BITRVI    X061,X188,512
BITRVI    X062,X124,512
BITRVI    X063,X252,512
BITRVI    X065,X130,512
BITRVI    X067,X194,512
BITRVI    X069,X162,512
BITRVI    X070,X098,512
BITRVI    X071,X226,512
BITRVI    X073,X146,512
BITRVI    X074,X082,512
BITRVI    X075,X210,512
BITRVI    X077,X178,512
BITRVI    X078,X114,512
BITRVI    X079,X242,512
BITRVI    X081,X138,512
BITRVI    X083,X202,512
BITRVI    X085,X170,512
BITRVI    X086,X106,512
BITRVI    X087,X234,512
BITRVI    X089,X154,512
BITRVI    X091,X218,512
BITRVI    X093,X186,512
BITRVI    X094,X122,512
BITRVI    X095,X250,512
BITRVI    X097,X134,512
BITRVI    X099,X198,512
BITRVI    X101,X166,512
BITRVI    X103,X230,512
BITRVI    X105,X150,512
BITRVI    X107,X214,512
BITRVI    X109,X182,512
BITRVI    X110,X118,512
BITRVI    X111,X246,512
BITRVI    X113,X142,512
BITRVI    X115,X206,512
BITRVI    X117,X174,512
BITRVI    X119,X238,512
BITRVI    X121,X158,512
BITRVI    X123,X222,512
BITRVI    X125,X190,512
BITRVI    X127,X254,512
BITRVI    X131,X193,512
BITRVI    X133,X161,512
BITRVI    X135,X225,512
BITRVI    X137,X145,512
BITRVI    X139,X209,512
BITRVI    X141,X177,512
BITRVI    X143,X241,512
BITRVI    X147,X201,512
BITRVI    X149,X169,512
BITRVI    X151,X233,512
BITRVI    X155,X217,512
BITRVI    X157,X185,512
BITRVI    X159,X249,512
BITRVI    X163,X197,512
BITRVI    X167,X229,512
BITRVI    X171,X213,512
BITRVI    X173,X181,512
BITRVI    X175,X245,512
BITRVI    X179,X205,512
BITRVI    X183,X237,512
BITRVI    X187,X221,512
BITRVI    X191,X253,512
BITRVI    X199,X227,512
BITRVI    X203,X211,512
BITRVI    X207,X243,512
BITRVI    X215,X235,512
BITRVI    X223,X251,512
BITRVI    X239,X247,512
*
*         FFT CODE FOR STAGES 1 AND 2
*
LDPK      4
COMBOI    X000,X001,X002,X003,0
COMBOI    X004,X005,X006,X007,0
COMBOI    X008,X009,X010,X011,0
COMBOI    X012,X013,X014,X015,0
COMBOI    X016,X017,X018,X019,0
COMBOI    X020,X021,X022,X023,0
COMBOI    X024,X025,X026,X027,0
COMBOI    X028,X029,X030,X031,0
COMBOI    X032,X033,X034,X035,0
```

```
         COMBOI    X036,X037,X038,X039,0
         COMBOI    X040,X041,X042,X043,0
         COMBOI    X044,X045,X046,X047,0
         COMBOI    X048,X049,X050,X051,0
         COMBOI    X052,X053,X054,X055,0
         COMBOI    X056,X057,X058,X059,0
         COMBOI    X060,X061,X062,X063,0
         LDPK      5
         COMBOI    X064,X065,X066,X067,128
         COMBOI    X068,X069,X070,X071,128
         COMBOI    X072,X073,X074,X075,128
         COMBOI    X076,X077,X078,X079,128
         COMBOI    X080,X081,X082,X083,128
         COMBOI    X084,X085,X086,X087,128
         COMBOI    X088,X089,X090,X091,128
         COMBOI    X092,X093,X094,X095,128
         COMBOI    X096,X097,X098,X099,128
         COMBOI    X100,X101,X102,X103,128
         COMBOI    X104,X105,X106,X107,128
         COMBOI    X108,X109,X110,X111,128
         COMBOI    X112,X113,X114,X115,128
         COMBOI    X116,X117,X118,X119,128
         COMBOI    X120,X121,X122,X123,128
         COMBOI    X124,X125,X126,X127,128
         LDPK      6
         COMBOI    X128,X129,X130,X131,256
         COMBOI    X132,X133,X134,X135,256
         COMBOI    X136,X137,X138,X139,256
         COMBOI    X140,X141,X142,X143,256
         COMBOI    X144,X145,X146,X147,256
         COMBOI    X148,X149,X150,X151,256
         COMBOI    X152,X153,X154,X155,256
         COMBOI    X156,X157,X158,X159,256
         COMBOI    X160,X161,X162,X163,256
         COMBOI    X164,X165,X166,X167,256
         COMBOI    X168,X169,X170,X171,256
         COMBOI    X172,X173,X174,X175,256
         COMBOI    X176,X177,X178,X179,256
         COMBOI    X180,X181,X182,X183,256
         COMBOI    X184,X185,X186,X187,256
         COMBOI    X188,X189,X190,X191,256
         LDPK      7
         COMBOI    X192,X193,X194,X195,384
         COMBOI    X196,X197,X198,X199,384
         COMBOI    X200,X201,X202,X203,384
         COMBOI    X204,X205,X206,X207,384
         COMBOI    X208,X209,X210,X211,384
         COMBOI    X212,X213,X214,X215,384
         COMBOI    X216,X217,X218,X219,384
         COMBOI    X220,X221,X222,X223,384
         COMBOI    X224,X225,X226,X227,384
         COMBOI    X228,X229,X230,X231,384
         COMBOI    X232,X233,X234,X235,384
         COMBOI    X236,X237,X238,X239,384
         COMBOI    X240,X241,X242,X243,384
         COMBOI    X244,X245,X246,X247,384
         COMBOI    X248,X249,X250,X251,384
         COMBOI    X252,X253,X254,X255,384
*
*        FFT CODE FOR STAGE 3
*
         ZEROI     X000,X004,512
         PBY4I     X001,X005,512
         PBY2I     X002,X006,512
         P3BY4I    X003,X007,512
         ZEROI     X008,X012,512
         PBY4I     X009,X013,512
         PBY2I     X010,X014,512
         P3BY4I    X011,X015,512
         ZEROI     X016,X020,512
         PBY4I     X017,X021,512
         PBY2I     X018,X022,512
         P3BY4I    X019,X023,512
         ZEROI     X024,X028,512
         PBY4I     X025,X029,512
         PBY2I     X026,X030,512
         P3BY4I    X027,X031,512
         ZEROI     X032,X036,512
         PBY4I     X033,X037,512
         PBY2I     X034,X038,512
         P3BY4I    X035,X039,512
         ZEROI     X040,X044,512
         PBY4I     X041,X045,512
         PBY2I     X042,X046,512
         P3BY4I    X043,X047,512
         ZEROI     X048,X052,512
         PBY4I     X049,X053,512
         PBY2I     X050,X054,512
         P3BY4I    X051,X055,512
         ZEROI     X056,X060,512
         PBY4I     X057,X061,512
         PBY2I     X058,X062,512
         P3BY4I    X059,X063,512
         ZEROI     X064,X068,512
         PBY4I     X065,X069,512
         PBY2I     X066,X070,512
         P3BY4I    X067,X071,512
         ZEROI     X072,X076,512
         PBY4I     X073,X077,512
         PBY2I     X074,X078,512
         P3BY4I    X075,X079,512
         ZEROI     X080,X084,512
         PBY4I     X081,X085,512
         PBY2I     X082,X086,512
         P3BY4I    X083,X087,512
         ZEROI     X088,X092,512
         PBY4I     X089,X093,512
         PBY2I     X090,X094,512
         P3BY4I    X091,X095,512
         ZEROI     X096,X100,512
         PBY4I     X097,X101,512
         PBY2I     X098,X102,512
         P3BY4I    X099,X103,512
         ZEROI     X104,X108,512
         PBY4I     X105,X109,512
         PBY2I     X106,X110,512
         P3BY4I    X107,X111,512
         ZEROI     X112,X116,512
         PBY4I     X113,X117,512
         PBY2I     X114,X118,512
```

```
P3BY4I    X115,X119,512
ZEROI     X120,X124,512
PBY4I     X121,X125,512
PBY2I     X122,X126,512
P3BY4I    X123,X127,512
ZEROI     X128,X132,512
PBY4I     X129,X133,512
PBY2I     X130,X134,512
P3BY4I    X131,X135,512
ZEROI     X136,X140,512
PBY4I     X137,X141,512
PBY2I     X138,X142,512
P3BY4I    X139,X143,512
ZEROI     X144,X148,512
PBY4I     X145,X149,512
PBY2I     X146,X150,512
P3BY4I    X147,X151,512
ZEROI     X152,X156,512
PBY4I     X153,X157,512
PBY2I     X154,X158,512
P3BY4I    X155,X159,512
ZEROI     X160,X164,512
PBY4I     X161,X165,512
PBY2I     X162,X166,512
P3BY4I    X163,X167,512
ZEROI     X168,X172,512
PBY4I     X169,X173,512
PBY2I     X170,X174,512
P3BY4I    X171,X175,512
ZEROI     X176,X180,512
PBY4I     X177,X181,512
PBY2I     X178,X182,512
P3BY4I    X179,X183,512
ZEROI     X184,X188,512
PBY4I     X185,X189,512
PBY2I     X186,X190,512
P3BY4I    X187,X191,512
ZEROI     X192,X196,512
PBY4I     X193,X197,512
PBY2I     X194,X198,512
P3BY4I    X195,X199,512
ZEROI     X200,X204,512
PBY4I     X201,X205,512
PBY2I     X202,X206,512
P3BY4I    X203,X207,512
ZEROI     X208,X212,512
PBY4I     X209,X213,512
PBY2I     X210,X214,512
P3BY4I    X211,X215,512
ZEROI     X216,X220,512
PBY4I     X217,X221,512
PBY2I     X218,X222,512
P3BY4I    X219,X223,512
ZEROI     X224,X228,512
PBY4I     X225,X229,512
PBY2I     X226,X230,512
P3BY4I    X227,X231,512
ZEROI     X232,X236,512
PBY4I     X233,X237,512
PBY2I     X234,X238,512
P3BY4I    X235,X239,512
ZEROI     X240,X244,512
PBY4I     X241,X245,512
PBY2I     X242,X246,512
P3BY4I    X243,X247,512
ZEROI     X248,X252,512
PBY4I     X249,X253,512
PBY2I     X250,X254,512
P3BY4I    X251,X255,512
*
*         FFT CODE FOR STAGE 4
*
ZEROI     X000,X008,512
BTRFLI    X001,X009,C016,S016,512
PBY4I     X002,X010,512
BTRFLI    X003,X011,C048,S048,512
PBY2I     X004,X012,512
BTRFLI    X005,X013,C080,S080,512
P3BY4I    X006,X014,512
BTRFLI    X007,X015,C112,S112,512
ZEROI     X016,X024,512
BTRFLI    X017,X025,C016,S016,512
PBY4I     X018,X026,512
BTRFLI    X019,X027,C048,S048,512
PBY2I     X020,X028,512
BTRFLI    X021,X029,C080,S080,512
P3BY4I    X022,X030,512
BTRFLI    X023,X031,C112,S112,512
ZEROI     X032,X040,512
BTRFLI    X033,X041,C016,S016,512
PBY4I     X034,X042,512
BTRFLI    X035,X043,C048,S048,512
PBY2I     X036,X044,512
BTRFLI    X037,X045,C080,S080,512
P3BY4I    X038,X046,512
BTRFLI    X039,X047,C112,S112,512
ZEROI     X048,X056,512
BTRFLI    X049,X057,C016,S016,512
PBY4I     X050,X058,512
BTRFLI    X051,X059,C048,S048,512
PBY2I     X052,X060,512
BTRFLI    X053,X061,C080,S080,512
P3BY4I    X054,X062,512
BTRFLI    X055,X063,C112,S112,512
ZEROI     X064,X072,512
BTRFLI    X065,X073,C016,S016,512
PBY4I     X066,X074,512
BTRFLI    X067,X075,C048,S048,512
PBY2I     X068,X076,512
BTRFLI    X069,X077,C080,S080,512
P3BY4I    X070,X078,512
BTRFLI    X071,X079,C112,S112,512
ZEROI     X080,X088,512
BTRFLI    X081,X089,C016,S016,512
PBY4I     X082,X090,512
BTRFLI    X083,X091,C048,S048,512
PBY2I     X084,X092,512
BTRFLI    X085,X093,C080,S080,512
P3BY4I    X086,X094,512
BTRFLI    X087,X095,C112,S112,512
```

```
         P3BY4I   X115,X119,512
         ZEROI    X120,X124,512
         PBY4I    X121,X125,512
         PBY2I    X122,X126,512
         P3BY4I   X123,X127,512
         ZEROI    X128,X132,512
         PBY4I    X129,X133,512
         PBY2I    X130,X134,512
         P3BY4I   X131,X135,512
         ZEROI    X136,X140,512
         PBY4I    X137,X141,512
         PBY2I    X138,X142,512
         P3BY4I   X139,X143,512
         ZEROI    X144,X148,512
         PBY4I    X145,X149,512
         PBY2I    X146,X150,512
         P3BY4I   X147,X151,512
         ZEROI    X152,X156,512
         PBY4I    X153,X157,512
         PBY2I    X154,X158,512
         P3BY4I   X155,X159,512
         ZEROI    X160,X164,512
         PBY4I    X161,X165,512
         PBY2I    X162,X166,512
         P3BY4I   X163,X167,512
         ZEROI    X168,X172,512
         PBY4I    X169,X173,512
         PBY2I    X170,X174,512
         P3BY4I   X171,X175,512
         ZEROI    X176,X180,512
         PBY4I    X177,X181,512
         PBY2I    X178,X182,512
         P3BY4I   X179,X183,512
         ZEROI    X184,X188,512
         PBY4I    X185,X189,512
         PBY2I    X186,X190,512
         P3BY4I   X187,X191,512
         ZEROI    X192,X196,512
         PBY4I    X193,X197,512
         PBY2I    X194,X198,512
         P3BY4I   X195,X199,512
         ZEROI    X200,X204,512
         PBY4I    X201,X205,512
         PBY2I    X202,X206,512
         P3BY4I   X203,X207,512
         ZEROI    X208,X212,512
         PBY4I    X209,X213,512
         PBY2I    X210,X214,512
         P3BY4I   X211,X215,512
         ZEROI    X216,X220,512
         PBY4I    X217,X221,512
         PBY2I    X218,X222,512
         P3BY4I   X219,X223,512
         ZEROI    X224,X228,512
         PBY4I    X225,X229,512
         PBY2I    X226,X230,512
         P3BY4I   X227,X231,512
         ZEROI    X232,X236,512
         PBY4I    X233,X237,512
         PBY2I    X234,X238,512
         P3BY4I   X235,X239,512
         ZEROI    X240,X244,512
         PBY4I    X241,X245,512
         PBY2I    X242,X246,512
         P3BY4I   X243,X247,512
         ZEROI    X248,X252,512
         PBY4I    X249,X253,512
         PBY2I    X250,X254,512
         P3BY4I   X251,X255,512
*
*        FFT CODE FOR STAGE 4
*
         ZEROI    X000,X008,512
         BTRFLI   X001,X009,C016,S016,512
         PBY4I    X002,X010,512
         BTRFLI   X003,X011,C048,S048,512
         PBY2I    X004,X012,512
         BTRFLI   X005,X013,C080,S080,512
         P3BY4I   X006,X014,512
         BTRFLI   X007,X015,C112,S112,512
         ZEROI    X016,X024,512
         BTRFLI   X017,X025,C016,S016,512
         PBY4I    X018,X026,512
         BTRFLI   X019,X027,C048,S048,512
         PBY2I    X020,X028,512
         BTRFLI   X021,X029,C080,S080,512
         P3BY4I   X022,X030,512
         BTRFLI   X023,X031,C112,S112,512
         ZEROI    X032,X040,512
         BTRFLI   X033,X041,C016,S016,512
         PBY4I    X034,X042,512
         BTRFLI   X035,X043,C048,S048,512
         PBY2I    X036,X044,512
         BTRFLI   X037,X045,C080,S080,512
         P3BY4I   X038,X046,512
         BTRFLI   X039,X047,C112,S112,512
         ZEROI    X048,X056,512
         BTRFLI   X049,X057,C016,S016,512
         PBY4I    X050,X058,512
         BTRFLI   X051,X059,C048,S048,512
         PBY2I    X052,X060,512
         BTRFLI   X053,X061,C080,S080,512
         P3BY4I   X054,X062,512
         BTRFLI   X055,X063,C112,S112,512
         ZEROI    X064,X072,512
         BTRFLI   X065,X073,C016,S016,512
         PBY4I    X066,X074,512
         BTRFLI   X067,X075,C048,S048,512
         PBY2I    X068,X076,512
         BTRFLI   X069,X077,C080,S080,512
         P3BY4I   X070,X078,512
         BTRFLI   X071,X079,C112,S112,512
         ZEROI    X080,X088,512
         BTRFLI   X081,X089,C016,S016,512
         PBY4I    X082,X090,512
         BTRFLI   X083,X091,C048,S048,512
         PBY2I    X084,X092,512
         BTRFLI   X085,X093,C080,S080,512
         P3BY4I   X086,X094,512
         BTRFLI   X087,X095,C112,S112,512
```

```
ZEROI    X096,X104,512
BTRFLI   X097,X105,C016,S016,512
PBY4I    X098,X106,512
BTRFLI   X099,X107,C048,S048,512
PBY2I    X100,X108,512
BTRFLI   X101,X109,C080,S080,512
P3BY4I   X102,X110,512
BTRFLI   X103,X111,C112,S112,512
ZEROI    X112,X120,512
BTRFLI   X113,X121,C016,S016,512
PBY4I    X114,X122,512
BTRFLI   X115,X123,C048,S048,512
PBY2I    X116,X124,512
BTRFLI   X117,X125,C080,S080,512
P3BY4I   X118,X126,512
BTRFLI   X119,X127,C112,S112,512
ZEROI    X128,X136,512
BTRFLI   X129,X137,C016,S016,512
PBY4I    X130,X138,512
BTRFLI   X131,X139,C048,S048,512
PBY2I    X132,X140,512
BTRFLI   X133,X141,C080,S080,512
P3BY4I   X134,X142,512
BTRFLI   X135,X143,C112,S112,512
ZEROI    X144,X152,512
BTRFLI   X145,X153,C016,S016,512
PBY4I    X146,X154,512
BTRFLI   X147,X155,C048,S048,512
PBY2I    X148,X156,512
BTRFLI   X149,X157,C080,S080,512
P3BY4I   X150,X158,512
BTRFLI   X151,X159,C112,S112,512
ZEROI    X160,X168,512
BTRFLI   X161,X169,C016,S016,512
PBY4I    X162,X170,512
BTRFLI   X163,X171,C048,S048,512
PBY2I    X164,X172,512
BTRFLI   X165,X173,C080,S080,512
P3BY4I   X166,X174,512
BTRFLI   X167,X175,C112,S112,512
ZEROI    X176,X184,512
BTRFLI   X177,X185,C016,S016,512
PBY4I    X178,X186,512
BTRFLI   X179,X187,C048,S048,512
PBY2I    X180,X188,512
BTRFLI   X181,X189,C080,S080,512
P3BY4I   X182,X190,512
BTRFLI   X183,X191,C112,S112,512
ZEROI    X192,X200,512
BTRFLI   X193,X201,C016,S016,512
PBY4I    X194,X202,512
BTRFLI   X195,X203,C048,S048,512
PBY2I    X196,X204,512
BTRFLI   X197,X205,C080,S080,512
P3BY4I   X198,X206,512
BTRFLI   X199,X207,C112,S112,512
ZEROI    X208,X216,512
BTRFLI   X209,X217,C016,S016,512
PBY4I    X210,X218,512
BTRFLI   X211,X219,C048,S048,512
PBY2I    X212,X220,512
BTRFLI   X213,X221,C080,S080,512
P3BY4I   X214,X222,512
BTRFLI   X215,X223,C112,S112,512
ZEROI    X224,X232,512
BTRFLI   X225,X233,C016,S016,512
PBY4I    X226,X234,512
BTRFLI   X227,X235,C048,S048,512
PBY2I    X228,X236,512
BTRFLI   X229,X237,C080,S080,512
P3BY4I   X230,X238,512
BTRFLI   X231,X239,C112,S112,512
ZEROI    X240,X248,512
BTRFLI   X241,X249,C016,S016,512
PBY4I    X242,X250,512
BTRFLI   X243,X251,C048,S048,512
PBY2I    X244,X252,512
BTRFLI   X245,X253,C080,S080,512
P3BY4I   X246,X254,512
BTRFLI   X247,X255,C112,S112,512
*
*        FFT CODE FOR STAGE 5
*
ZEROI    X000,X016,512
BTRFLI   X001,X017,C008,S008,512
BTRFLI   X002,X018,C016,S016,512
BTRFLI   X003,X019,C024,S024,512
PBY4I    X004,X020,512
BTRFLI   X005,X021,C040,S040,512
BTRFLI   X006,X022,C048,S048,512
BTRFLI   X007,X023,C056,S056,512
PBY2I    X008,X024,512
BTRFLI   X009,X025,C072,S072,512
BTRFLI   X010,X026,C080,S080,512
BTRFLI   X011,X027,C088,S088,512
P3BY4I   X012,X028,512
BTRFLI   X013,X029,C104,S104,512
BTRFLI   X014,X030,C112,S112,512
BTRFLI   X015,X031,C120,S120,512
ZEROI    X032,X048,512
BTRFLI   X033,X049,C008,S008,512
BTRFLI   X034,X050,C016,S016,512
BTRFLI   X035,X051,C024,S024,512
PBY4I    X036,X052,512
BTRFLI   X037,X053,C040,S040,512
BTRFLI   X038,X054,C048,S048,512
BTRFLI   X039,X055,C056,S056,512
PBY2I    X040,X056,512
BTRFLI   X041,X057,C072,S072,512
BTRFLI   X042,X058,C080,S080,512
BTRFLI   X043,X059,C088,S088,512
P3BY4I   X044,X060,512
BTRFLI   X045,X061,C104,S104,512
BTRFLI   X046,X062,C112,S112,512
BTRFLI   X047,X063,C120,S120,512
ZEROI    X064,X080,512
BTRFLI   X065,X081,C008,S008,512
BTRFLI   X066,X082,C016,S016,512
BTRFLI   X067,X083,C024,S024,512
PBY4I    X068,X084,512
```

```
BTRFLI   X069,X085,C040,S040,512
BTRFLI   X070,X086,C048,S048,512
BTRFLI   X071,X087,C056,S056,512
PBY2I    X072,X088,512
BTRFLI   X073,X089,C072,S072,512
BTRFLI   X074,X090,C080,S080,512
BTRFLI   X075,X091,C088,S088,512
P3BY4I   X076,X092,512
BTRFLI   X077,X093,C104,S104,512
BTRFLI   X078,X094,C112,S112,512
BTRFLI   X079,X095,C120,S120,512
ZEROI    X096,X112,512
BTRFLI   X097,X113,C008,S008,512
BTRFLI   X098,X114,C016,S016,512
BTRFLI   X099,X115,C024,S024,512
PBY4I    X100,X116,512
BTRFLI   X101,X117,C040,S040,512
BTRFLI   X102,X118,C048,S048,512
BTRFLI   X103,X119,C056,S056,512
PBY2I    X104,X120,512
BTRFLI   X105,X121,C072,S072,512
BTRFLI   X106,X122,C080,S080,512
BTRFLI   X107,X123,C088,S088,512
P3BY4I   X108,X124,512
BTRFLI   X109,X125,C104,S104,512
BTRFLI   X110,X126,C112,S112,512
BTRFLI   X111,X127,C120,S120,512
ZEROI    X128,X144,512
BTRFLI   X129,X145,C008,S008,512
BTRFLI   X130,X146,C016,S016,512
BTRFLI   X131,X147,C024,S024,512
PBY4I    X132,X148,512
BTRFLI   X133,X149,C040,S040,512
BTRFLI   X134,X150,C048,S048,512
BTRFLI   X135,X151,C056,S056,512
PBY2I    X136,X152,512
BTRFLI   X137,X153,C072,S072,512
BTRFLI   X138,X154,C080,S080,512
BTRFLI   X139,X155,C088,S088,512
P3BY4I   X140,X156,512
BTRFLI   X141,X157,C104,S104,512
BTRFLI   X142,X158,C112,S112,512
BTRFLI   X143,X159,C120,S120,512
ZEROI    X160,X176,512
BTRFLI   X161,X177,C008,S008,512
BTRFLI   X162,X178,C016,S016,512
BTRFLI   X163,X179,C024,S024,512
PBY4I    X164,X180,512
BTRFLI   X165,X181,C040,S040,512
BTRFLI   X166,X182,C048,S048,512
BTRFLI   X167,X183,C056,S056,512
PBY2I    X168,X184,512
BTRFLI   X169,X185,C072,S072,512
BTRFLI   X170,X186,C080,S080,512
BTRFLI   X171,X187,C088,S088,512
P3BY4I   X172,X188,512
BTRFLI   X173,X189,C104,S104,512
BTRFLI   X174,X190,C112,S112,512
BTRFLI   X175,X191,C120,S120,512
ZEROI    X192,X208,512
BTRFLI   X193,X209,C008,S008,512
BTRFLI   X194,X210,C016,S016,512
BTRFLI   X195,X211,C024,S024,512
PBY4I    X196,X212,512
BTRFLI   X197,X213,C040,S040,512
BTRFLI   X198,X214,C048,S048,512
BTRFLI   X199,X215,C056,S056,512
PBY2I    X200,X216,512
BTRFLI   X201,X217,C072,S072,512
BTRFLI   X202,X218,C080,S080,512
BTRFLI   X203,X219,C088,S088,512
P3BY4I   X204,X220,512
BTRFLI   X205,X221,C104,S104,512
BTRFLI   X206,X222,C112,S112,512
BTRFLI   X207,X223,C120,S120,512
ZEROI    X224,X240,512
BTRFLI   X225,X241,C008,S008,512
BTRFLI   X226,X242,C016,S016,512
BTRFLI   X227,X243,C024,S024,512
PBY4I    X228,X244,512
BTRFLI   X229,X245,C040,S040,512
BTRFLI   X230,X246,C048,S048,512
BTRFLI   X231,X247,C056,S056,512
PBY2I    X232,X248,512
BTRFLI   X233,X249,C072,S072,512
BTRFLI   X234,X250,C080,S080,512
BTRFLI   X235,X251,C088,S088,512
P3BY4I   X236,X252,512
BTRFLI   X237,X253,C104,S104,512
BTRFLI   X238,X254,C112,S112,512
BTRFLI   X239,X255,C120,S120,512
*
*        FFT CODE FOR STAGE 6
*
ZEROI    X000,X032,512
BTRFLI   X001,X033,C004,S004,512
BTRFLI   X002,X034,C008,S008,512
BTRFLI   X003,X035,C012,S012,512
BTRFLI   X004,X036,C016,S016,512
BTRFLI   X005,X037,C020,S020,512
BTRFLI   X006,X038,C024,S024,512
BTRFLI   X007,X039,C028,S028,512
PBY4I    X008,X040,512
BTRFLI   X009,X041,C036,S036,512
BTRFLI   X010,X042,C040,S040,512
BTRFLI   X011,X043,C044,S044,512
BTRFLI   X012,X044,C048,S048,512
BTRFLI   X013,X045,C052,S052,512
BTRFLI   X014,X046,C056,S056,512
BTRFLI   X015,X047,C060,S060,512
PBY2I    X016,X048,512
BTRFLI   X017,X049,C068,S068,512
BTRFLI   X018,X050,C072,S072,512
BTRFLI   X019,X051,C076,S076,512
BTRFLI   X020,X052,C080,S080,512
BTRFLI   X021,X053,C084,S084,512
BTRFLI   X022,X054,C088,S088,512
BTRFLI   X023,X055,C092,S092,512
P3BY4I   X024,X056,512
BTRFLI   X025,X057,C100,S100,512
```

```
BTRFLI    X026,X058,C104,S104,512
BTRFLI    X027,X059,C108,S108,512
BTRFLI    X028,X060,C112,S112,512
BTRFLI    X029,X061,C116,S116,512
BTRFLI    X030,X062,C120,S120,512
BTRFLI    X031,X063,C124,S124,512
ZEROI     X064,X096,512
BTRFLI    X065,X097,C004,S004,512
BTRFLI    X066,X098,C008,S008,512
BTRFLI    X067,X099,C012,S012,512
BTRFLI    X068,X100,C016,S016,512
BTRFLI    X069,X101,C020,S020,512
BTRFLI    X070,X102,C024,S024,512
BTRFLI    X071,X103,C028,S028,512
PBY4I     X072,X104,512
BTRFLI    X073,X105,C036,S036,512
BTRFLI    X074,X106,C040,S040,512
BTRFLI    X075,X107,C044,S044,512
BTRFLI    X076,X108,C048,S048,512
BTRFLI    X077,X109,C052,S052,512
BTRFLI    X078,X110,C056,S056,512
BTRFLI    X079,X111,C060,S060,512
PBY2I     X080,X112,512
BTRFLI    X081,X113,C068,S068,512
BTRFLI    X082,X114,C072,S072,512
BTRFLI    X083,X115,C076,S076,512
BTRFLI    X084,X116,C080,S080,512
BTRFLI    X085,X117,C084,S084,512
BTRFLI    X086,X118,C088,S088,512
BTRFLI    X087,X119,C092,S092,512
P3BY4I    X088,X120,512
BTRFLI    X089,X121,C100,S100,512
BTRFLI    X090,X122,C104,S104,512
BTRFLI    X091,X123,C108,S108,512
BTRFLI    X092,X124,C112,S112,512
BTRFLI    X093,X125,C116,S116,512
BTRFLI    X094,X126,C120,S120,512
BTRFLI    X095,X127,C124,S124,512
ZEROI     X128,X160,512
BTRFLI    X129,X161,C004,S004,512
BTRFLI    X130,X162,C008,S008,512
BTRFLI    X131,X163,C012,S012,512
BTRFLI    X132,X164,C016,S016,512
BTRFLI    X133,X165,C020,S020,512
BTRFLI    X134,X166,C024,S024,512
BTRFLI    X135,X167,C028,S028,512
PBY4I     X136,X168,512
BTRFLI    X137,X169,C036,S036,512
BTRFLI    X138,X170,C040,S040,512
BTRFLI    X139,X171,C044,S044,512
BTRFLI    X140,X172,C048,S048,512
BTRFLI    X141,X173,C052,S052,512
BTRFLI    X142,X174,C056,S056,512
BTRFLI    X143,X175,C060,S060,512
PBY2I     X144,X176,512
BTRFLI    X145,X177,C068,S068,512
BTRFLI    X146,X178,C072,S072,512
BTRFLI    X147,X179,C076,S076,512
BTRFLI    X148,X180,C080,S080,512
BTRFLI    X149,X181,C084,S084,512
BTRFLI    X150,X182,C088,S088,512
BTRFLI    X151,X183,C092,S092,512
P3BY4I    X152,X184,512
BTRFLI    X153,X185,C100,S100,512
BTRFLI    X154,X186,C104,S104,512
BTRFLI    X155,X187,C108,S108,512
BTRFLI    X156,X188,C112,S112,512
BTRFLI    X157,X189,C116,S116,512
BTRFLI    X158,X190,C120,S120,512
BTRFLI    X159,X191,C124,S124,512
ZEROI     X192,X224,512
BTRFLI    X193,X225,C004,S004,512
BTRFLI    X194,X226,C008,S008,512
BTRFLI    X195,X227,C012,S012,512
BTRFLI    X196,X228,C016,S016,512
BTRFLI    X197,X229,C020,S020,512
BTRFLI    X198,X230,C024,S024,512
BTRFLI    X199,X231,C028,S028,512
PBY4I     X200,X232,512
BTRFLI    X201,X233,C036,S036,512
BTRFLI    X202,X234,C040,S040,512
BTRFLI    X203,X235,C044,S044,512
BTRFLI    X204,X236,C048,S048,512
BTRFLI    X205,X237,C052,S052,512
BTRFLI    X206,X238,C056,S056,512
BTRFLI    X207,X239,C060,S060,512
PBY2I     X208,X240,512
BTRFLI    X209,X241,C068,S068,512
BTRFLI    X210,X242,C072,S072,512
BTRFLI    X211,X243,C076,S076,512
BTRFLI    X212,X244,C080,S080,512
BTRFLI    X213,X245,C084,S084,512
BTRFLI    X214,X246,C088,S088,512
BTRFLI    X215,X247,C092,S092,512
P3BY4I    X216,X248,512
BTRFLI    X217,X249,C100,S100,512
BTRFLI    X218,X250,C104,S104,512
BTRFLI    X219,X251,C108,S108,512
BTRFLI    X220,X252,C112,S112,512
BTRFLI    X221,X253,C116,S116,512
BTRFLI    X222,X254,C120,S120,512
BTRFLI    X223,X255,C124,S124,512
*
*         FFT CODE FOR STAGE 7
*
ZEROI     X000,X064,512
BTRFLI    X001,X065,C002,S002,512
BTRFLI    X002,X066,C004,S004,512
BTRFLI    X003,X067,C006,S006,512
BTRFLI    X004,X068,C008,S008,512
BTRFLI    X005,X069,C010,S010,512
BTRFLI    X006,X070,C012,S012,512
BTRFLI    X007,X071,C014,S014,512
BTRFLI    X008,X072,C016,S016,512
BTRFLI    X009,X073,C018,S018,512
BTRFLI    X010,X074,C020,S020,512
BTRFLI    X011,X075,C022,S022,512
BTRFLI    X012,X076,C024,S024,512
BTRFLI    X013,X077,C026,S026,512
BTRFLI    X014,X078,C028,S028,512
```

```
BTRFLI    X015,X079,C030,S030,512
PBY4I     X016,X080,512
BTRFLI    X017,X081,C034,S034,512
BTRFLI    X018,X082,C036,S036,512
BTRFLI    X019,X083,C038,S038,512
BTRFLI    X020,X084,C040,S040,512
BTRFLI    X021,X085,C042,S042,512
BTRFLI    X022,X086,C044,S044,512
BTRFLI    X023,X087,C046,S046,512
BTRFLI    X024,X088,C048,S048,512
BTRFLI    X025,X089,C050,S050,512
BTRFLI    X026,X090,C052,S052,512
BTRFLI    X027,X091,C054,S054,512
BTRFLI    X028,X092,C056,S056,512
BTRFLI    X029,X093,C058,S058,512
BTRFLI    X030,X094,C060,S060,512
BTRFLI    X031,X095,C062,S062,512
PBY2I     X032,X096,512
BTRFLI    X033,X097,C066,S066,512
BTRFLI    X034,X098,C068,S068,512
BTRFLI    X035,X099,C070,S070,512
BTRFLI    X036,X100,C072,S072,512
BTRFLI    X037,X101,C074,S074,512
BTRFLI    X038,X102,C076,S076,512
BTRFLI    X039,X103,C078,S078,512
BTRFLI    X040,X104,C080,S080,512
BTRFLI    X041,X105,C082,S082,512
BTRFLI    X042,X106,C084,S084,512
BTRFLI    X043,X107,C086,S086,512
BTRFLI    X044,X108,C088,S088,512
BTRFLI    X045,X109,C090,S090,512
BTRFLI    X046,X110,C092,S092,512
BTRFLI    X047,X111,C094,S094,512
P3BY4I    X048,X112,512
BTRFLI    X049,X113,C098,S098,512
BTRFLI    X050,X114,C100,S100,512
BTRFLI    X051,X115,C102,S102,512
BTRFLI    X052,X116,C104,S104,512
BTRFLI    X053,X117,C106,S106,512
BTRFLI    X054,X118,C108,S108,512
BTRFLI    X055,X119,C110,S110,512
BTRFLI    X056,X120,C112,S112,512
BTRFLI    X057,X121,C114,S114,512
BTRFLI    X058,X122,C116,S116,512
BTRFLI    X059,X123,C118,S118,512
BTRFLI    X060,X124,C120,S120,512
BTRFLI    X061,X125,C122,S122,512
BTRFLI    X062,X126,C124,S124,512
BTRFLI    X063,X127,C126,S126,512
ZEROI     X128,X192,512
BTRFLI    X129,X193,C002,S002,512
BTRFLI    X130,X194,C004,S004,512
BTRFLI    X131,X195,C006,S006,512
BTRFLI    X132,X196,C008,S008,512
BTRFLI    X133,X197,C010,S010,512
BTRFLI    X134,X198,C012,S012,512
BTRFLI    X135,X199,C014,S014,512
BTRFLI    X136,X200,C016,S016,512
BTRFLI    X137,X201,C018,S018,512
BTRFLI    X138,X202,C020,S020,512
BTRFLI    X139,X203,C022,S022,512
BTRFLI    X140,X204,C024,S024,512
BTRFLI    X141,X205,C026,S026,512
BTRFLI    X142,X206,C028,S028,512
BTRFLI    X143,X207,C030,S030,512
PBY4I     X144,X208,512
BTRFLI    X145,X209,C034,S034,512
BTRFLI    X146,X210,C036,S036,512
BTRFLI    X147,X211,C038,S038,512
BTRFLI    X148,X212,C040,S040,512
BTRFLI    X149,X213,C042,S042,512
BTRFLI    X150,X214,C044,S044,512
BTRFLI    X151,X215,C046,S046,512
BTRFLI    X152,X216,C048,S048,512
BTRFLI    X153,X217,C050,S050,512
BTRFLI    X154,X218,C052,S052,512
BTRFLI    X155,X219,C054,S054,512
BTRFLI    X156,X220,C056,S056,512
BTRFLI    X157,X221,C058,S058,512
BTRFLI    X158,X222,C060,S060,512
BTRFLI    X159,X223,C062,S062,512
PBY2I     X160,X224,512
BTRFLI    X161,X225,C066,S066,512
BTRFLI    X162,X226,C068,S068,512
BTRFLI    X163,X227,C070,S070,512
BTRFLI    X164,X228,C072,S072,512
BTRFLI    X165,X229,C074,S074,512
BTRFLI    X166,X230,C076,S076,512
BTRFLI    X167,X231,C078,S078,512
BTRFLI    X168,X232,C080,S080,512
BTRFLI    X169,X233,C082,S082,512
BTRFLI    X170,X234,C084,S084,512
BTRFLI    X171,X235,C086,S086,512
BTRFLI    X172,X236,C088,S088,512
BTRFLI    X173,X237,C090,S090,512
BTRFLI    X174,X238,C092,S092,512
BTRFLI    X175,X239,C094,S094,512
P3BY4I    X176,X240,512
BTRFLI    X177,X241,C098,S098,512
BTRFLI    X178,X242,C100,S100,512
BTRFLI    X179,X243,C102,S102,512
BTRFLI    X180,X244,C104,S104,512
BTRFLI    X181,X245,C106,S106,512
BTRFLI    X182,X246,C108,S108,512
BTRFLI    X183,X247,C110,S110,512
BTRFLI    X184,X248,C112,S112,512
BTRFLI    X185,X249,C114,S114,512
BTRFLI    X186,X250,C116,S116,512
BTRFLI    X187,X251,C118,S118,512
BTRFLI    X188,X252,C120,S120,512
BTRFLI    X189,X253,C122,S122,512
BTRFLI    X190,X254,C124,S124,512
BTRFLI    X191,X255,C126,S126,512
*
*         FFT CODE FOR STAGE 8
*
ZEROI     X000,X128,512
BTRFLI    X001,X129,C001,S001,512
BTRFLI    X002,X130,C002,S002,512
BTRFLI    X003,X131,C003,S003,512
```

```
BTRFLI    X004,X132,C004,S004,512
BTRFLI    X005,X133,C005,S005,512
BTRFLI    X006,X134,C006,S006,512
BTRFLI    X007,X135,C007,S007,512
BTRFLI    X008,X136,C008,S008,512
BTRFLI    X009,X137,C009,S009,512
BTRFLI    X010,X138,C010,S010,512
BTRFLI    X011,X139,C011,S011,512
BTRFLI    X012,X140,C012,S012,512
BTRFLI    X013,X141,C013,S013,512
BTRFLI    X014,X142,C014,S014,512
BTRFLI    X015,X143,C015,S015,512
BTRFLI    X016,X144,C016,S016,512
BTRFLI    X017,X145,C017,S017,512
BTRFLI    X018,X146,C018,S018,512
BTRFLI    X019,X147,C019,S019,512
BTRFLI    X020,X148,C020,S020,512
BTRFLI    X021,X149,C021,S021,512
BTRFLI    X022,X150,C022,S022,512
BTRFLI    X023,X151,C023,S023,512
BTRFLI    X024,X152,C024,S024,512
BTRFLI    X025,X153,C025,S025,512
BTRFLI    X026,X154,C026,S026,512
BTRFLI    X027,X155,C027,S027,512
BTRFLI    X028,X156,C028,S028,512
BTRFLI    X029,X157,C029,S029,512
BTRFLI    X030,X158,C030,S030,512
BTRFLI    X031,X159,C031,S031,512
PBY4I     X032,X160,512
BTRFLI    X033,X161,C033,S033,512
BTRFLI    X034,X162,C034,S034,512
BTRFLI    X035,X163,C035,S035,512
BTRFLI    X036,X164,C036,S036,512
BTRFLI    X037,X165,C037,S037,512
BTRFLI    X038,X166,C038,S038,512
BTRFLI    X039,X167,C039,S039,512
BTRFLI    X040,X168,C040,S040,512
BTRFLI    X041,X169,C041,S041,512
BTRFLI    X042,X170,C042,S042,512
BTRFLI    X043,X171,C043,S043,512
BTRFLI    X044,X172,C044,S044,512
BTRFLI    X045,X173,C045,S045,512
BTRFLI    X046,X174,C046,S046,512
BTRFLI    X047,X175,C047,S047,512
BTRFLI    X048,X176,C048,S048,512
BTRFLI    X049,X177,C049,S049,512
BTRFLI    X050,X178,C050,S050,512
BTRFLI    X051,X179,C051,S051,512
BTRFLI    X052,X180,C052,S052,512
BTRFLI    X053,X181,C053,S053,512
BTRFLI    X054,X182,C054,S054,512
BTRFLI    X055,X183,C055,S055,512
BTRFLI    X056,X184,C056,S056,512
BTRFLI    X057,X185,C057,S057,512
BTRFLI    X058,X186,C058,S058,512
BTRFLI    X059,X187,C059,S059,512
BTRFLI    X060,X188,C060,S060,512
BTRFLI    X061,X189,C061,S061,512
BTRFLI    X062,X190,C062,S062,512
BTRFLI    X063,X191,C063,S063,512
PBY2I     X064,X192,512
BTRFLI    X065,X193,C065,S065,512
BTRFLI    X066,X194,C066,S066,512
BTRFLI    X067,X195,C067,S067,512
BTRFLI    X068,X196,C068,S068,512
BTRFLI    X069,X197,C069,S069,512
BTRFLI    X070,X198,C070,S070,512
BTRFLI    X071,X199,C071,S071,512
BTRFLI    X072,X200,C072,S072,512
BTRFLI    X073,X201,C073,S073,512
BTRFLI    X074,X202,C074,S074,512
BTRFLI    X075,X203,C075,S075,512
BTRFLI    X076,X204,C076,S076,512
BTRFLI    X077,X205,C077,S077,512
BTRFLI    X078,X206,C078,S078,512
BTRFLI    X079,X207,C079,S079,512
BTRFLI    X080,X208,C080,S080,512
BTRFLI    X081,X209,C081,S081,512
BTRFLI    X082,X210,C082,S082,512
BTRFLI    X083,X211,C083,S083,512
BTRFLI    X084,X212,C084,S084,512
BTRFLI    X085,X213,C085,S085,512
BTRFLI    X086,X214,C086,S086,512
BTRFLI    X087,X215,C087,S087,512
BTRFLI    X088,X216,C088,S088,512
BTRFLI    X089,X217,C089,S089,512
BTRFLI    X090,X218,C090,S090,512
BTRFLI    X091,X219,C091,S091,512
BTRFLI    X092,X220,C092,S092,512
BTRFLI    X093,X221,C093,S093,512
BTRFLI    X094,X222,C094,S094,512
BTRFLI    X095,X223,C095,S095,512
P3BY4I    X096,X224,512
BTRFLI    X097,X225,C097,S097,512
BTRFLI    X098,X226,C098,S098,512
BTRFLI    X099,X227,C099,S099,512
BTRFLI    X100,X228,C100,S100,512
BTRFLI    X101,X229,C101,S101,512
BTRFLI    X102,X230,C102,S102,512
BTRFLI    X103,X231,C103,S103,512
BTRFLI    X104,X232,C104,S104,512
BTRFLI    X105,X233,C105,S105,512
BTRFLI    X106,X234,C106,S106,512
BTRFLI    X107,X235,C107,S107,512
BTRFLI    X108,X236,C108,S108,512
BTRFLI    X109,X237,C109,S109,512
BTRFLI    X110,X238,C110,S110,512
BTRFLI    X111,X239,C111,S111,512
BTRFLI    X112,X240,C112,S112,512
BTRFLI    X113,X241,C113,S113,512
BTRFLI    X114,X242,C114,S114,512
BTRFLI    X115,X243,C115,S115,512
BTRFLI    X116,X244,C116,S116,512
BTRFLI    X117,X245,C117,S117,512
BTRFLI    X118,X246,C118,S118,512
BTRFLI    X119,X247,C119,S119,512
BTRFLI    X120,X248,C120,S120,512
BTRFLI    X121,X249,C121,S121,512
BTRFLI    X122,X250,C122,S122,512
BTRFLI    X123,X251,C123,S123,512
BTRFLI    X124,X252,C124,S124,512
BTRFLI    X125,X253,C125,S125,512
BTRFLI    X126,X254,C126,S126,512
BTRFLI    X127,X255,C127,S127,512
END
```

APPENDIX D
A 1024-POINT, RADIX-2 DIT FFT IMPLEMENTATION

```
          A P P E N D I X   D
          -------------------

          IDT       'FT1024'
*************************************************************
*                                                           *
*   A 1024-POINT RADIX-2 DIT COMPLEX FFT FOR THE TMS32020   *
*   ----------------------------------------------------    *
*                                                           *
*   THE FOLLOWING FILE RAD2FFT.MAC CONSISTS OF ALL THE      *
*   MACROS LISTED IN APPENDIX B                             *
*                                                           *
*************************************************************
*
          COPY      RAD2FFT.MAC
*
*************************************************************
*                                                           *
*   DATA MEMORY MAP FOR PAGES 4, 5, 6 AND 7 (BLOCKS B0,B1)  *
*                                                           *
*************************************************************
*
          DORG    0
*
*         DATA MEMORY PAGE 4 (STARTING ADDRESS 512 OR >200)
*
X000      DATA    0,0
X001      DATA    0,0
X002      DATA    0,0
X003      DATA    0,0
X004      DATA    0,0
X005      DATA    0,0
X006      DATA    0,0
X007      DATA    0,0
X008      DATA    0,0
X009      DATA    0,0
X010      DATA    0,0
X011      DATA    0,0
X012      DATA    0,0
X013      DATA    0,0
X014      DATA    0,0
X015      DATA    0,0
X016      DATA    0,0
X017      DATA    0,0
X018      DATA    0,0
X019      DATA    0,0
X020      DATA    0,0
X021      DATA    0,0
X022      DATA    0,0
X023      DATA    0,0
X024      DATA    0,0
X025      DATA    0,0
X026      DATA    0,0
X027      DATA    0,0
X028      DATA    0,0
X029      DATA    0,0
X030      DATA    0,0
X031      DATA    0,0
X032      DATA    0,0
X033      DATA    0,0
```

```
X034      DATA    0,0
X035      DATA    0,0
X036      DATA    0,0
X037      DATA    0,0
X038      DATA    0,0
X039      DATA    0,0
X040      DATA    0,0
X041      DATA    0,0
X042      DATA    0,0
X043      DATA    0,0
X044      DATA    0,0
X045      DATA    0,0
X046      DATA    0,0
X047      DATA    0,0
X048      DATA    0,0
X049      DATA    0,0
X050      DATA    0,0
X051      DATA    0,0
X052      DATA    0,0
X053      DATA    0,0
X054      DATA    0,0
X055      DATA    0,0
X056      DATA    0,0
X057      DATA    0,0
X058      DATA    0,0
X059      DATA    0,0
X060      DATA    0,0
X061      DATA    0,0
X062      DATA    0,0
X063      DATA    0,0
*
*         DATA MEMORY PAGE 5 (STARTING ADDRESS 640 OR >280)
*
X064      DATA    0,0
X065      DATA    0,0
X066      DATA    0,0
X067      DATA    0,0
X068      DATA    0,0
X069      DATA    0,0
X070      DATA    0,0
X071      DATA    0,0
X072      DATA    0,0
X073      DATA    0,0
X074      DATA    0,0
X075      DATA    0,0
X076      DATA    0,0
X077      DATA    0,0
X078      DATA    0,0
X079      DATA    0,0
X080      DATA    0,0
X081      DATA    0,0
X082      DATA    0,0
X083      DATA    0,0
X084      DATA    0,0
X085      DATA    0,0
X086      DATA    0,0
X087      DATA    0,0
X088      DATA    0,0
X089      DATA    0,0
X090      DATA    0,0
X091      DATA    0,0
X092      DATA    0,0
X093      DATA    0,0
X094      DATA    0,0
X095      DATA    0,0
X096      DATA    0,0
X097      DATA    0,0
X098      DATA    0,0
X099      DATA    0,0
X100      DATA    0,0
X101      DATA    0,0
X102      DATA    0,0
X103      DATA    0,0
X104      DATA    0,0
X105      DATA    0,0
X106      DATA    0,0
X107      DATA    0,0
X108      DATA    0,0
X109      DATA    0,0
X110      DATA    0,0
X111      DATA    0,0
X112      DATA    0,0
X113      DATA    0,0
X114      DATA    0,0
X115      DATA    0,0
X116      DATA    0,0
X117      DATA    0,0
X118      DATA    0,0
X119      DATA    0,0
X120      DATA    0,0
X121      DATA    0,0
X122      DATA    0,0
X123      DATA    0,0
X124      DATA    0,0
X125      DATA    0,0
X126      DATA    0,0
X127      DATA    0,0
*
*         DATA MEMORY PAGE 6 (STARTING ADDRESS 768 OR >300)
*
X128      DATA    0,0
X129      DATA    0,0
X130      DATA    0,0
X131      DATA    0,0
X132      DATA    0,0
X133      DATA    0,0
X134      DATA    0,0
X135      DATA    0,0
X136      DATA    0,0
X137      DATA    0,0
X138      DATA    0,0
X139      DATA    0,0
X140      DATA    0,0
X141      DATA    0,0
X142      DATA    0,0
X143      DATA    0,0
X144      DATA    0,0
X145      DATA    0,0
X146      DATA    0,0
X147      DATA    0,0
```

```
X148      DATA     0,0
X149      DATA     0,0
X150      DATA     0,0
X151      DATA     0,0
X152      DATA     0,0
X153      DATA     0,0
X154      DATA     0,0
X155      DATA     0,0
X156      DATA     0,0
X157      DATA     0,0
X158      DATA     0,0
X159      DATA     0,0
X160      DATA     0,0
X161      DATA     0,0
X162      DATA     0,0
X163      DATA     0,0
X164      DATA     0,0
X165      DATA     0,0
X166      DATA     0,0
X167      DATA     0,0
X168      DATA     0,0
X169      DATA     0,0
X170      DATA     0,0
X171      DATA     0,0
X172      DATA     0,0
X173      DATA     0,0
X174      DATA     0,0
X175      DATA     0,0
X176      DATA     0,0
X177      DATA     0,0
X178      DATA     0,0
X179      DATA     0,0
X180      DATA     0,0
X181      DATA     0,0
X182      DATA     0,0
X183      DATA     0,0
X184      DATA     0,0
X185      DATA     0,0
X186      DATA     0,0
X187      DATA     0,0
X188      DATA     0,0
X189      DATA     0,0
X190      DATA     0,0
X191      DATA     0,0
*
*         DATA MEMORY PAGE 7 (STARTING ADDRESS 896 OR >380)
*
X192      DATA     0,0
X193      DATA     0,0
X194      DATA     0,0
X195      DATA     0,0
X196      DATA     0,0
X197      DATA     0,0
X198      DATA     0,0
X199      DATA     0,0
X200      DATA     0,0
X201      DATA     0,0
X202      DATA     0,0
X203      DATA     0,0
X204      DATA     0,0
X205      DATA     0,0
X206      DATA     0,0
X207      DATA     0,0
X208      DATA     0,0
X209      DATA     0,0
X210      DATA     0,0
X211      DATA     0,0
X212      DATA     0,0
X213      DATA     0,0
X214      DATA     0,0
X215      DATA     0,0
X216      DATA     0,0
X217      DATA     0,0
X218      DATA     0,0
X219      DATA     0,0
X220      DATA     0,0
X221      DATA     0,0
X222      DATA     0,0
X223      DATA     0,0
X224      DATA     0,0
X225      DATA     0,0
X226      DATA     0,0
X227      DATA     0,0
X228      DATA     0,0
X229      DATA     0,0
X230      DATA     0,0
X231      DATA     0,0
X232      DATA     0,0
X233      DATA     0,0
X234      DATA     0,0
X235      DATA     0,0
X236      DATA     0,0
X237      DATA     0,0
X238      DATA     0,0
X239      DATA     0,0
X240      DATA     0,0
X241      DATA     0,0
X242      DATA     0,0
X243      DATA     0,0
X244      DATA     0,0
X245      DATA     0,0
X246      DATA     0,0
X247      DATA     0,0
X248      DATA     0,0
X249      DATA     0,0
X250      DATA     0,0
X251      DATA     0,0
X252      DATA     0,0
X253      DATA     0,0
X254      DATA     0,0
X255      DATA     0,0
*
**************************************************************
*                                                            *
*    DATA LOCATION IN BLOCK B2 FOR W=COS(45) OR SIN(45)      *
*                                                            *
**************************************************************
*
          DORG     96
W         DATA     0
```

```
*
****************************************************************
*                                                              *
*   13-BIT TWIDDLE FACTORS FOR 256-POINT COMPLEX FFT           *
*                                                              *
****************************************************************
*
C000    EQU     4095
C001    EQU     4094
C002    EQU     4091
C003    EQU     4085
C004    EQU     4076
C005    EQU     4065
C006    EQU     4052
C007    EQU     4036
C008    EQU     4017
C009    EQU     3996
C010    EQU     3973
C011    EQU     3948
C012    EQU     3920
C013    EQU     3889
C014    EQU     3857
C015    EQU     3822
C016    EQU     3784
C017    EQU     3745
C018    EQU     3703
C019    EQU     3659
C020    EQU     3612
C021    EQU     3564
C022    EQU     3513
C023    EQU     3461
C024    EQU     3406
C025    EQU     3349
C026    EQU     3290
C027    EQU     3229
C028    EQU     3166
C029    EQU     3102
C030    EQU     3035
C031    EQU     2967
C032    EQU     2896
C033    EQU     2824
C034    EQU     2751
C035    EQU     2675
C036    EQU     2598
C037    EQU     2520
C038    EQU     2440
C039    EQU     2359
C040    EQU     2276
C041    EQU     2191
C042    EQU     2106
C043    EQU     2019
C044    EQU     1931
C045    EQU     1842
C046    EQU     1751
C047    EQU     1660
C048    EQU     1567
C049    EQU     1474
C050    EQU     1380
C051    EQU     1285
C052    EQU     1189
C053    EQU     1092
C054    EQU     995
C055    EQU     897
C056    EQU     799
C057    EQU     700
C058    EQU     601
C059    EQU     501
C060    EQU     401
C061    EQU     301
C062    EQU     201
C063    EQU     101
C064    EQU     0
C065    EQU     -101
C066    EQU     -201
C067    EQU     -301
C068    EQU     -401
C069    EQU     -501
C070    EQU     -601
C071    EQU     -700
C072    EQU     -799
C073    EQU     -897
C074    EQU     -995
C075    EQU     -1092
C076    EQU     -1189
C077    EQU     -1285
C078    EQU     -1380
C079    EQU     -1474
C080    EQU     -1567
C081    EQU     -1660
C082    EQU     -1751
C083    EQU     -1842
C084    EQU     -1931
C085    EQU     -2019
C086    EQU     -2106
C087    EQU     -2191
C088    EQU     -2276
C089    EQU     -2359
C090    EQU     -2440
C091    EQU     -2520
C092    EQU     -2598
C093    EQU     -2675
C094    EQU     -2751
C095    EQU     -2824
C096    EQU     -2896
C097    EQU     -2967
C098    EQU     -3035
C099    EQU     -3102
C100    EQU     -3166
C101    EQU     -3229
C102    EQU     -3290
C103    EQU     -3349
C104    EQU     -3406
C105    EQU     -3461
C106    EQU     -3513
C107    EQU     -3564
C108    EQU     -3612
C109    EQU     -3659
C110    EQU     -3703
C111    EQU     -3745
C112    EQU     -3784
```

```
C113     EQU     -3822
C114     EQU     -3857
C115     EQU     -3889
C116     EQU     -3920
C117     EQU     -3948
C118     EQU     -3973
C119     EQU     -3996
C120     EQU     -4017
C121     EQU     -4036
C122     EQU     -4052
C123     EQU     -4065
C124     EQU     -4076
C125     EQU     -4085
C126     EQU     -4091
C127     EQU     -4094
*
S000     EQU     0
S001     EQU     101
S002     EQU     201
S003     EQU     301
S004     EQU     401
S005     EQU     501
S006     EQU     601
S007     EQU     700
S008     EQU     799
S009     EQU     897
S010     EQU     995
S011     EQU     1092
S012     EQU     1189
S013     EQU     1285
S014     EQU     1380
S015     EQU     1474
S016     EQU     1567
S017     EQU     1660
S018     EQU     1751
S019     EQU     1842
S020     EQU     1931
S021     EQU     2019
S022     EQU     2106
S023     EQU     2191
S024     EQU     2276
S025     EQU     2359
S026     EQU     2440
S027     EQU     2520
S028     EQU     2598
S029     EQU     2675
S030     EQU     2751
S031     EQU     2824
S032     EQU     2896
S033     EQU     2967
S034     EQU     3035
S035     EQU     3102
S036     EQU     3166
S037     EQU     3229
S038     EQU     3290
S039     EQU     3349
S040     EQU     3406
S041     EQU     3461
S042     EQU     3513
S043     EQU     3564
S044     EQU     3612
S045     EQU     3659
S046     EQU     3703
S047     EQU     3745
S048     EQU     3784
S049     EQU     3822
S050     EQU     3857
S051     EQU     3889
S052     EQU     3920
S053     EQU     3948
S054     EQU     3973
S055     EQU     3996
S056     EQU     4017
S057     EQU     4036
S058     EQU     4052
S059     EQU     4065
S060     EQU     4076
S061     EQU     4085
S062     EQU     4091
S063     EQU     4094
S064     EQU     4095
S065     EQU     4094
S066     EQU     4091
S067     EQU     4085
S068     EQU     4076
S069     EQU     4065
S070     EQU     4052
S071     EQU     4036
S072     EQU     4017
S073     EQU     3996
S074     EQU     3973
S075     EQU     3948
S076     EQU     3920
S077     EQU     3889
S078     EQU     3857
S079     EQU     3822
S080     EQU     3784
S081     EQU     3745
S082     EQU     3703
S083     EQU     3659
S084     EQU     3612
S085     EQU     3564
S086     EQU     3513
S087     EQU     3461
S088     EQU     3406
S089     EQU     3349
S090     EQU     3290
S091     EQU     3229
S092     EQU     3166
S093     EQU     3102
S094     EQU     3035
S095     EQU     2967
S096     EQU     2896
S097     EQU     2824
S098     EQU     2751
S099     EQU     2675
S100     EQU     2598
S101     EQU     2520
S102     EQU     2440
S103     EQU     2359
```

```
S104      EQU       2276
S105      EQU       2191
S106      EQU       2106
S107      EQU       2019
S108      EQU       1931
S109      EQU       1842
S110      EQU       1751
S111      EQU       1660
S112      EQU       1567
S113      EQU       1474
S114      EQU       1380
S115      EQU       1285
S116      EQU       1189
S117      EQU       1092
S118      EQU       995
S119      EQU       897
S120      EQU       799
S121      EQU       700
S122      EQU       601
S123      EQU       501
S124      EQU       401
S125      EQU       301
S126      EQU       201
S127      EQU       101
*
          LIST
          AORG      0
          B         INIT
*
          AORG      >20
*
WVAL      DATA      >5A82
*
*         SYSTEM INITIALIZATION
*
INIT      SPM       0
          CNFD
          ROVM
          SSXM
          LARP      AR0
          LRLK      AR0,W
          LALK      WVAL
          TBLR      *,AR1
          B         FT1024
*
*************************************************************
*                                                           *
*         256-POINT FFT KERNEL - STAGES 1 AND 2             *
*                                                           *
*************************************************************
*
KNL256    LDPK      4
          COMBOI    X000,X001,X002,X003,0
          COMBOI    X004,X005,X006,X007,0
          COMBOI    X008,X009,X010,X011,0
          COMBOI    X012,X013,X014,X015,0
          COMBOI    X016,X017,X018,X019,0
          COMBOI    X020,X021,X022,X023,0
          COMBOI    X024,X025,X026,X027,0
          COMBOI    X028,X029,X030,X031,0
          COMBOI    X032,X033,X034,X035,0
          COMBOI    X036,X037,X038,X039,0
          COMBOI    X040,X041,X042,X043,0
          COMBOI    X044,X045,X046,X047,0
          COMBOI    X048,X049,X050,X051,0
          COMBOI    X052,X053,X054,X055,0
          COMBOI    X056,X057,X058,X059,0
          COMBOI    X060,X061,X062,X063,0
          LDPK      5
          COMBOI    X064,X065,X066,X067,128
          COMBOI    X068,X069,X070,X071,128
          COMBOI    X072,X073,X074,X075,128
          COMBOI    X076,X077,X078,X079,128
          COMBOI    X080,X081,X082,X083,128
          COMBOI    X084,X085,X086,X087,128
          COMBOI    X088,X089,X090,X091,128
          COMBOI    X092,X093,X094,X095,128
          COMBOI    X096,X097,X098,X099,128
          COMBOI    X100,X101,X102,X103,128
          COMBOI    X104,X105,X106,X107,128
          COMBOI    X108,X109,X110,X111,128
          COMBOI    X112,X113,X114,X115,128
          COMBOI    X116,X117,X118,X119,128
          COMBOI    X120,X121,X122,X123,128
          COMBOI    X124,X125,X126,X127,128
          LDPK      6
          COMBOI    X128,X129,X130,X131,256
          COMBOI    X132,X133,X134,X135,256
          COMBOI    X136,X137,X138,X139,256
          COMBOI    X140,X141,X142,X143,256
          COMBOI    X144,X145,X146,X147,256
          COMBOI    X148,X149,X150,X151,256
          COMBOI    X152,X153,X154,X155,256
          COMBOI    X156,X157,X158,X159,256
          COMBOI    X160,X161,X162,X163,256
          COMBOI    X164,X165,X166,X167,256
          COMBOI    X168,X169,X170,X171,256
          COMBOI    X172,X173,X174,X175,256
          COMBOI    X176,X177,X178,X179,256
          COMBOI    X180,X181,X182,X183,256
          COMBOI    X184,X185,X186,X187,256
          COMBOI    X188,X189,X190,X191,256
          LDPK      7
          COMBOI    X192,X193,X194,X195,384
          COMBOI    X196,X197,X198,X199,384
          COMBOI    X200,X201,X202,X203,384
          COMBOI    X204,X205,X206,X207,384
          COMBOI    X208,X209,X210,X211,384
          COMBOI    X212,X213,X214,X215,384
          COMBOI    X216,X217,X218,X219,384
          COMBOI    X220,X221,X222,X223,384
          COMBOI    X224,X225,X226,X227,384
          COMBOI    X228,X229,X230,X231,384
          COMBOI    X232,X233,X234,X235,384
          COMBOI    X236,X237,X238,X239,384
          COMBOI    X240,X241,X242,X243,384
          COMBOI    X244,X245,X246,X247,384
          COMBOI    X248,X249,X250,X251,384
          COMBOI    X252,X253,X254,X255,384
*
```

```
*****************************************************************
*                                                               *
*         256-POINT FFT KERNEL - STAGE 3                        *
*                                                               *
*****************************************************************
*
          ZEROI     X000,X004,512
          PBY4I     X001,X005,512
          PBY2I     X002,X006,512
          P3BY4I    X003,X007,512
          ZEROI     X008,X012,512
          PBY4I     X009,X013,512
          PBY2I     X010,X014,512
          P3BY4I    X011,X015,512
          ZEROI     X016,X020,512
          PBY4I     X017,X021,512
          PBY2I     X018,X022,512
          P3BY4I    X019,X023,512
          ZEROI     X024,X028,512
          PBY4I     X025,X029,512
          PBY2I     X026,X030,512
          P3BY4I    X027,X031,512
          ZEROI     X032,X036,512
          PBY4I     X033,X037,512
          PBY2I     X034,X038,512
          P3BY4I    X035,X039,512
          ZEROI     X040,X044,512
          PBY4I     X041,X045,512
          PBY2I     X042,X046,512
          P3BY4I    X043,X047,512
          ZEROI     X048,X052,512
          PBY4I     X049,X053,512
          PBY2I     X050,X054,512
          P3BY4I    X051,X055,512
          ZEROI     X056,X060,512
          PBY4I     X057,X061,512
          PBY2I     X058,X062,512
          P3BY4I    X059,X063,512
          ZEROI     X064,X068,512
          PBY4I     X065,X069,512
          PBY2I     X066,X070,512
          P3BY4I    X067,X071,512
          ZEROI     X072,X076,512
          PBY4I     X073,X077,512
          PBY2I     X074,X078,512
          P3BY4I    X075,X079,512
          ZEROI     X080,X084,512
          PBY4I     X081,X085,512
          PBY2I     X082,X086,512
          P3BY4I    X083,X087,512
          ZEROI     X088,X092,512
          PBY4I     X089,X093,512
          PBY2I     X090,X094,512
          P3BY4I    X091,X095,512
          ZEROI     X096,X100,512
          PBY4I     X097,X101,512
          PBY2I     X098,X102,512
          P3BY4I    X099,X103,512
          ZEROI     X104,X108,512
          PBY4I     X105,X109,512
          PBY2I     X106,X110,512
          P3BY4I    X107,X111,512
          ZEROI     X112,X116,512
          PBY4I     X113,X117,512
          PBY2I     X114,X118,512
          P3BY4I    X115,X119,512
          ZEROI     X120,X124,512
          PBY4I     X121,X125,512
          PBY2I     X122,X126,512
          P3BY4I    X123,X127,512
          ZEROI     X128,X132,512
          PBY4I     X129,X133,512
          PBY2I     X130,X134,512
          P3BY4I    X131,X135,512
          ZEROI     X136,X140,512
          PBY4I     X137,X141,512
          PBY2I     X138,X142,512
          P3BY4I    X139,X143,512
          ZEROI     X144,X148,512
          PBY4I     X145,X149,512
          PBY2I     X146,X150,512
          P3BY4I    X147,X151,512
          ZEROI     X152,X156,512
          PBY4I     X153,X157,512
          PBY2I     X154,X158,512
          P3BY4I    X155,X159,512
          ZEROI     X160,X164,512
          PBY4I     X161,X165,512
          PBY2I     X162,X166,512
          P3BY4I    X163,X167,512
          ZEROI     X168,X172,512
          PBY4I     X169,X173,512
          PBY2I     X170,X174,512
          P3BY4I    X171,X175,512
          ZEROI     X176,X180,512
          PBY4I     X177,X181,512
          PBY2I     X178,X182,512
          P3BY4I    X179,X183,512
          ZEROI     X184,X188,512
          PBY4I     X185,X189,512
          PBY2I     X186,X190,512
          P3BY4I    X187,X191,512
          ZEROI     X192,X196,512
          PBY4I     X193,X197,512
          PBY2I     X194,X198,512
          P3BY4I    X195,X199,512
          ZEROI     X200,X204,512
          PBY4I     X201,X205,512
          PBY2I     X202,X206,512
          P3BY4I    X203,X207,512
          ZEROI     X208,X212,512
          PBY4I     X209,X213,512
          PBY2I     X210,X214,512
          P3BY4I    X211,X215,512
          ZEROI     X216,X220,512
          PBY4I     X217,X221,512
          PBY2I     X218,X222,512
          P3BY4I    X219,X223,512
          ZEROI     X224,X228,512
          PBY4I     X225,X229,512
```

```
         PBY2I     X226,X230,512
         P3BY4I    X227,X231,512
         ZEROI     X232,X236,512
         PBY4I     X233,X237,512
         PBY2I     X234,X238,512
         P3BY4I    X235,X239,512
         ZEROI     X240,X244,512
         PBY4I     X241,X245,512
         PBY2I     X242,X246,512
         P3BY4I    X243,X247,512
         ZEROI     X248,X252,512
         PBY4I     X249,X253,512
         PBY2I     X250,X254,512
         P3BY4I    X251,X255,512
*
*************************************************************
*                                                           *
*        256-POINT FFT KERNEL - STAGE 4                     *
*                                                           *
*************************************************************
*
         ZEROI     X000,X008,512
         BTRFLI    X001,X009,C016,S016,512
         PBY4I     X002,X010,512
         BTRFLI    X003,X011,C048,S048,512
         PBY2I     X004,X012,512
         BTRFLI    X005,X013,C080,S080,512
         P3BY4I    X006,X014,512
         BTRFLI    X007,X015,C112,S112,512
         ZEROI     X016,X024,512
         BTRFLI    X017,X025,C016,S016,512
         PBY4I     X018,X026,512
         BTRFLI    X019,X027,C048,S048,512
         PBY2I     X020,X028,512
         BTRFLI    X021,X029,C080,S080,512
         P3BY4I    X022,X030,512
         BTRFLI    X023,X031,C112,S112,512
         ZEROI     X032,X040,512
         BTRFLI    X033,X041,C016,S016,512
         PBY4I     X034,X042,512
         BTRFLI    X035,X043,C048,S048,512
         PBY2I     X036,X044,512
         BTRFLI    X037,X045,C080,S080,512
         P3BY4I    X038,X046,512
         BTRFLI    X039,X047,C112,S112,512
         ZEROI     X048,X056,512
         BTRFLI    X049,X057,C016,S016,512
         PBY4I     X050,X058,512
         BTRFLI    X051,X059,C048,S048,512
         PBY2I     X052,X060,512
         BTRFLI    X053,X061,C080,S080,512
         P3BY4I    X054,X062,512
         BTRFLI    X055,X063,C112,S112,512
         ZEROI     X064,X072,512
         BTRFLI    X065,X073,C016,S016,512
         PBY4I     X066,X074,512
         BTRFLI    X067,X075,C048,S048,512
         PBY2I     X068,X076,512
         BTRFLI    X069,X077,C080,S080,512
         P3BY4I    X070,X078,512
         BTRFLI    X071,X079,C112,S112,512
         ZEROI     X080,X088,512
         BTRFLI    X081,X089,C016,S016,512
         PBY4I     X082,X090,512
         BTRFLI    X083,X091,C048,S048,512
         PBY2I     X084,X092,512
         BTRFLI    X085,X093,C080,S080,512
         P3BY4I    X086,X094,512
         BTRFLI    X087,X095,C112,S112,512
         ZEROI     X096,X104,512
         BTRFLI    X097,X105,C016,S016,512
         PBY4I     X098,X106,512
         BTRFLI    X099,X107,C048,S048,512
         PBY2I     X100,X108,512
         BTRFLI    X101,X109,C080,S080,512
         P3BY4I    X102,X110,512
         BTRFLI    X103,X111,C112,S112,512
         ZEROI     X112,X120,512
         BTRFLI    X113,X121,C016,S016,512
         PBY4I     X114,X122,512
         BTRFLI    X115,X123,C048,S048,512
         PBY2I     X116,X124,512
         BTRFLI    X117,X125,C080,S080,512
         P3BY4I    X118,X126,512
         BTRFLI    X119,X127,C112,S112,512
         ZEROI     X128,X136,512
         BTRFLI    X129,X137,C016,S016,512
         PBY4I     X130,X138,512
         BTRFLI    X131,X139,C048,S048,512
         PBY2I     X132,X140,512
         BTRFLI    X133,X141,C080,S080,512
         P3BY4I    X134,X142,512
         BTRFLI    X135,X143,C112,S112,512
         ZEROI     X144,X152,512
         BTRFLI    X145,X153,C016,S016,512
         PBY4I     X146,X154,512
         BTRFLI    X147,X155,C048,S048,512
         PBY2I     X148,X156,512
         BTRFLI    X149,X157,C080,S080,512
         P3BY4I    X150,X158,512
         BTRFLI    X151,X159,C112,S112,512
         ZEROI     X160,X168,512
         BTRFLI    X161,X169,C016,S016,512
         PBY4I     X162,X170,512
         BTRFLI    X163,X171,C048,S048,512
         PBY2I     X164,X172,512
         BTRFLI    X165,X173,C080,S080,512
         P3BY4I    X166,X174,512
         BTRFLI    X167,X175,C112,S112,512
         ZEROI     X176,X184,512
         BTRFLI    X177,X185,C016,S016,512
         PBY4I     X178,X186,512
         BTRFLI    X179,X187,C048,S048,512
         PBY2I     X180,X188,512
         BTRFLI    X181,X189,C080,S080,512
         P3BY4I    X182,X190,512
         BTRFLI    X183,X191,C112,S112,512
         ZEROI     X192,X200,512
         BTRFLI    X193,X201,C016,S016,512
         PBY4I     X194,X202,512
```

```
         BTRFLI    X195,X203,C048,S048,512
         PBY2I     X196,X204,512
         BTRFLI    X197,X205,C080,S080,512
         P3BY4I    X198,X206,512
         BTRFLI    X199,X207,C112,S112,512
         ZEROI     X208,X216,512
         BTRFLI    X209,X217,C016,S016,512
         PBY4I     X210,X218,512
         BTRFLI    X211,X219,C048,S048,512
         PBY2I     X212,X220,512
         BTRFLI    X213,X221,C080,S080,512
         P3BY4I    X214,X222,512
         BTRFLI    X215,X223,C112,S112,512
         ZEROI     X224,X232,512
         BTRFLI    X225,X233,C016,S016,512
         PBY4I     X226,X234,512
         BTRFLI    X227,X235,C048,S048,512
         PBY2I     X228,X236,512
         BTRFLI    X229,X237,C080,S080,512
         P3BY4I    X230,X238,512
         BTRFLI    X231,X239,C112,S112,512
         ZEROI     X240,X248,512
         BTRFLI    X241,X249,C016,S016,512
         PBY4I     X242,X250,512
         BTRFLI    X243,X251,C048,S048,512
         PBY2I     X244,X252,512
         BTRFLI    X245,X253,C080,S080,512
         P3BY4I    X246,X254,512
         BTRFLI    X247,X255,C112,S112,512
*
*************************************************************
*                                                           *
*        256-POINT FFT KERNEL - STAGE 5                     *
*                                                           *
*************************************************************
*
         ZEROI     X000,X016,512
         BTRFLI    X001,X017,C008,S008,512
         BTRFLI    X002,X018,C016,S016,512
         BTRFLI    X003,X019,C024,S024,512
         PBY4I     X004,X020,512
         BTRFLI    X005,X021,C040,S040,512
         BTRFLI    X006,X022,C048,S048,512
         BTRFLI    X007,X023,C056,S056,512
         PBY2I     X008,X024,512
         BTRFLI    X009,X025,C072,S072,512
         BTRFLI    X010,X026,C080,S080,512
         BTRFLI    X011,X027,C088,S088,512
         P3BY4I    X012,X028,512
         BTRFLI    X013,X029,C104,S104,512
         BTRFLI    X014,X030,C112,S112,512
         BTRFLI    X015,X031,C120,S120,512
         ZEROI     X032,X048,512
         BTRFLI    X033,X049,C008,S008,512
         BTRFLI    X034,X050,C016,S016,512
         BTRFLI    X035,X051,C024,S024,512
         PBY4I     X036,X052,512
         BTRFLI    X037,X053,C040,S040,512
         BTRFLI    X038,X054,C048,S048,512
         BTRFLI    X039,X055,C056,S056,512
         PBY2I     X040,X056,512
         BTRFLI    X041,X057,C072,S072,512
         BTRFLI    X042,X058,C080,S080,512
         BTRFLI    X043,X059,C088,S088,512
         P3BY4I    X044,X060,512
         BTRFLI    X045,X061,C104,S104,512
         BTRFLI    X046,X062,C112,S112,512
         BTRFLI    X047,X063,C120,S120,512
         ZEROI     X064,X080,512
         BTRFLI    X065,X081,C008,S008,512
         BTRFLI    X066,X082,C016,S016,512
         BTRFLI    X067,X083,C024,S024,5_2
         PBY4I     X068,X084,512
         BTRFLI    X069,X085,C040,S040,512
         BTRFLI    X070,X086,C048,S048,512
         BTRFLI    X071,X087,C056,S056,512
         PBY2I     X072,X088,512
         BTRFLI    X073,X089,C072,S072,512
         BTRFLI    X074,X090,C080,S080,512
         BTRFLI    X075,X091,C088,S088,512
         P3BY4I    X076,X092,512
         BTRFLI    X077,X093,C104,S104,512
         BTRFLI    X078,X094,C112,S112,512
         BTRFLI    X079,X095,C120,S120,512
         ZEROI     X096,X112,512
         BTRFLI    X097,X113,C008,S008,512
         BTRFLI    X098,X114,C016,S016,512
         BTRFLI    X099,X115,C024,S024,512
         PBY4I     X100,X116,512
         BTRFLI    X101,X117,C040,S040,512
         BTRFLI    X102,X118,C048,S048,512
         BTRFLI    X103,X119,C056,S056,512
         PBY2I     X104,X120,512
         BTRFLI    X105,X121,C072,S072,512
         BTRFLI    X106,X122,C080,S080,512
         BTRFLI    X107,X123,C088,S088,512
         P3BY4I    X108,X124,512
         BTRFLI    X109,X125,C104,S104,512
         BTRFLI    X110,X126,C112,S112,512
         BTRFLI    X111,X127,C120,S120,512
         ZEROI     X128,X144,512
         BTRFLI    X129,X145,C008,S008,512
         BTRFLI    X130,X146,C016,S016,512
         BTRFLI    X131,X147,C024,S024,512
         PBY4I     X132,X148,512
         BTRFLI    X133,X149,C040,S040,512
         BTRFLI    X134,X150,C048,S048,512
         BTRFLI    X135,X151,C056,S056,512
         PBY2I     X136,X152,512
         BTRFLI    X137,X153,C072,S072,512
         BTRFLI    X138,X154,C080,S080,512
         BTRFLI    X139,X155,C088,S088,512
         P3BY4I    X140,X156,512
         BTRFLI    X141,X157,C104,S104,512
         BTRFLI    X142,X158,C112,S112,512
         BTRFLI    X143,X159,C120,S120,512
         ZEROI     X160,X176,512
         BTRFLI    X161,X177,C008,S008,512
         BTRFLI    X162,X178,C016,S016,512
         BTRFLI    X163,X179,C024,S024,512
```

```
         PBY4I     X164,X180,512
         BTRFLI    X165,X181,C040,S040,512
         BTRFLI    X166,X182,C048,S048,512
         BTRFLI    X167,X183,C056,S056,512
         PBY2I     X168,X184,512
         BTRFLI    X169,X185,C072,S072,512
         BTRFLI    X170,X186,C080,S080,512
         BTRFLI    X171,X187,C088,S088,512
         P3BY4I    X172,X188,512
         BTRFLI    X173,X189,C104,S104,512
         BTRFLI    X174,X190,C112,S112,512
         BTRFLI    X175,X191,C120,S120,512
         ZEROI     X192,X208,512
         BTRFLI    X193,X209,C008,S008,512
         BTRFLI    X194,X210,C016,S016,512
         BTRFLI    X195,X211,C024,S024,512
         PBY4I     X196,X212,512
         BTRFLI    X197,X213,C040,S040,512
         BTRFLI    X198,X214,C048,S048,512
         BTRFLI    X199,X215,C056,S056,512
         PBY2I     X200,X216,512
         BTRFLI    X201,X217,C072,S072,512
         BTRFLI    X202,X218,C080,S080,512
         BTRFLI    X203,X219,C088,S088,512
         P3BY4I    X204,X220,512
         BTRFLI    X205,X221,C104,S104,512
         BTRFLI    X206,X222,C112,S112,512
         BTRFLI    X207,X223,C120,S120,512
         ZEROI     X224,X240,512
         BTRFLI    X225,X241,C008,S008,512
         BTRFLI    X226,X242,C016,S016,512
         BTRFLI    X227,X243,C024,S024,512
         PBY4I     X228,X244,512
         BTRFLI    X229,X245,C040,S040,512
         BTRFLI    X230,X246,C048,S048,512
         BTRFLI    X231,X247,C056,S056,512
         PBY2I     X232,X248,512
         BTRFLI    X233,X249,C072,S072,512
         BTRFLI    X234,X250,C080,S080,512
         BTRFLI    X235,X251,C088,S088,512
         P3BY4I    X236,X252,512
         BTRFLI    X237,X253,C104,S104,512
         BTRFLI    X238,X254,C112,S112,512
         BTRFLI    X239,X255,C120,S120,512
*
*****************************************************************
*                                                               *
*        256-POINT FFT KERNEL - STAGE 6                         *
*                                                               *
*****************************************************************
*
         ZEROI     X000,X032,512
         BTRFLI    X001,X033,C004,S004,512
         BTRFLI    X002,X034,C008,S008,512
         BTRFLI    X003,X035,C012,S012,512
         BTRFLI    X004,X036,C016,S016,512
         BTRFLI    X005,X037,C020,S020,512
         BTRFLI    X006,X038,C024,S024,512
         BTRFLI    X007,X039,C028,S028,512
         PBY4I     X008,X040,512
         BTRFLI    X009,X041,C036,S036,512
         BTRFLI    X010,X042,C040,S040,512
         BTRFLI    X011,X043,C044,S044,512
         BTRFLI    X012,X044,C048,S048,512
         BTRFLI    X013,X045,C052,S052,512
         BTRFLI    X014,X046,C056,S056,512
         BTRFLI    X015,X047,C060,S060,512
         PBY2I     X016,X048,512
         BTRFLI    X017,X049,C068,S068,512
         BTRFLI    X018,X050,C072,S072,512
         BTRFLI    X019,X051,C076,S076,512
         BTRFLI    X020,X052,C080,S080,512
         BTRFLI    X021,X053,C084,S084,512
         BTRFLI    X022,X054,C088,S088,512
         BTRFLI    X023,X055,C092,S092,512
         P3BY4I    X024,X056,512
         BTRFLI    X025,X057,C100,S100,512
         BTRFLI    X026,X058,C104,S104,512
         BTRFLI    X027,X059,C108,S108,512
         BTRFLI    X028,X060,C112,S112,512
         BTRFLI    X029,X061,C116,S116,512
         BTRFLI    X030,X062,C120,S120,512
         BTRFLI    X031,X063,C124,S124,512
         ZEROI     X064,X096,512
         BTRFLI    X065,X097,C004,S004,512
         BTRFLI    X066,X098,C008,S008,512
         BTRFLI    X067,X099,C012,S012,512
         BTRFLI    X068,X100,C016,S016,512
         BTRFLI    X069,X101,C020,S020,512
         BTRFLI    X070,X102,C024,S024,512
         BTRFLI    X071,X103,C028,S028,512
         PBY4I     X072,X104,512
         BTRFLI    X073,X105,C036,S036,512
         BTRFLI    X074,X106,C040,S040,512
         BTRFLI    X075,X107,C044,S044,512
         BTRFLI    X076,X108,C048,S048,512
         BTRFLI    X077,X109,C052,S052,512
         BTRFLI    X078,X110,C056,S056,512
         BTRFLI    X079,X111,C060,S060,512
         PBY2I     X080,X112,512
         BTRFLI    X081,X113,C068,S068,512
         BTRFLI    X082,X114,C072,S072,512
         BTRFLI    X083,X115,C076,S076,512
         BTRFLI    X084,X116,C080,S080,512
         BTRFLI    X085,X117,C084,S084,512
         BTRFLI    X086,X118,C088,S088,512
         BTRFLI    X087,X119,C092,S092,512
         P3BY4I    X088,X120,512
         BTRFLI    X089,X121,C100,S100,512
         BTRFLI    X090,X122,C104,S104,512
         BTRFLI    X091,X123,C108,S108,512
         BTRFLI    X092,X124,C112,S112,512
         BTRFLI    X093,X125,C116,S116,512
         BTRFLI    X094,X126,C120,S120,512
         BTRFLI    X095,X127,C124,S124,512
         ZEROI     X128,X160,512
         BTRFLI    X129,X161,C004,S004,512
         BTRFLI    X130,X162,C008,S008,512
         BTRFLI    X131,X163,C012,S012,512
         BTRFLI    X132,X164,C016,S016,512
```

```
         BTRFLI    X133,X165,C020,S020,512
         BTRFLI    X134,X166,C024,S024,512
         BTRFLI    X135,X167,C028,S028,512
         PBY4I     X136,X168,512
         BTRFLI    X137,X169,C036,S036,512
         BTRFLI    X138,X170,C040,S040,512
         BTRFLI    X139,X171,C044,S044,512
         BTRFLI    X140,X172,C048,S048,512
         BTRFLI    X141,X173,C052,S052,512
         BTRFLI    X142,X174,C056,S056,512
         BTRFLI    X143,X175,C060,S060,512
         PBY2I     X144,X176,512
         BTRFLI    X145,X177,C068,S068,512
         BTRFLI    X146,X178,C072,S072,512
         BTRFLI    X147,X179,C076,S076,512
         BTRFLI    X148,X180,C080,S080,512
         BTRFLI    X149,X181,C084,S084,512
         BTRFLI    X150,X182,C088,S088,512
         BTRFLI    X151,X183,C092,S092,512
         P3BY4I    X152,X184,512
         BTRFLI    X153,X185,C100,S100,512
         BTRFLI    X154,X186,C104,S104,512
         BTRFLI    X155,X187,C108,S108,512
         BTRFLI    X156,X188,C112,S112,512
         BTRFLI    X157,X189,C116,S116,512
         BTRFLI    X158,X190,C120,S120,512
         BTRFLI    X159,X191,C124,S124,512
         ZEROI     X192,X224,512
         BTRFLI    X193,X225,C004,S004,512
         BTRFLI    X194,X226,C008,S008,512
         BTRFLI    X195,X227,C012,S012,512
         BTRFLI    X196,X228,C016,S016,512
         BTRFLI    X197,X229,C020,S020,512
         BTRFLI    X198,X230,C024,S024,512
         BTRFLI    X199,X231,C028,S028,512
         PBY4I     X200,X232,512
         BTRFLI    X201,X233,C036,S036,512
         BTRFLI    X202,X234,C040,S040,512
         BTRFLI    X203,X235,C044,S044,512
         BTRFLI    X204,X236,C048,S048,512
         BTRFLI    X205,X237,C052,S052,512
         BTRFLI    X206,X238,C056,S056,512
         BTRFLI    X207,X239,C060,S060,512
         PBY2I     X208,X240,512
         BTRFLI    X209,X241,C068,S068,512
         BTRFLI    X210,X242,C072,S072,512
         BTRFLI    X211,X243,C076,S076,512
         BTRFLI    X212,X244,C080,S080,512
         BTRFLI    X213,X245,C084,S084,512
         BTRFLI    X214,X246,C088,S088,512
         BTRFLI    X215,X247,C092,S092,512
         P3BY4I    X216,X248,512
         BTRFLI    X217,X249,C100,S100,512
         BTRFLI    X218,X250,C104,S104,512
         BTRFLI    X219,X251,C108,S108,512
         BTRFLI    X220,X252,C112,S112,512
         BTRFLI    X221,X253,C116,S116,512
         BTRFLI    X222,X254,C120,S120,512
         BTRFLI    X223,X255,C124,S124,512
*
***************************************************************
*                                                             *
*        256-POINT FFT KERNEL - STAGE 7                       *
*                                                             *
***************************************************************
*
         ZEROI     X000,X064,512
         BTRFLI    X001,X065,C002,S002,512
         BTRFLI    X002,X066,C004,S004,512
         BTRFLI    X003,X067,C006,S006,512
         BTRFLI    X004,X068,C008,S008,512
         BTRFLI    X005,X069,C010,S010,512
         BTRFLI    X006,X070,C012,S012,512
         BTRFLI    X007,X071,C014,S014,512
         BTRFLI    X008,X072,C016,S016,512
         BTRFLI    X009,X073,C018,S018,512
         BTRFLI    X010,X074,C020,S020,512
         BTRFLI    X011,X075,C022,S022,512
         BTRFLI    X012,X076,C024,S024,512
         BTRFLI    X013,X077,C026,S026,512
         BTRFLI    X014,X078,C028,S028,512
         BTRFLI    X015,X079,C030,S030,512
         PBY4I     X016,X080,512
         BTRFLI    X017,X081,C034,S034,512
         BTRFLI    X018,X082,C036,S036,512
         BTRFLI    X019,X083,C038,S038,512
         BTRFLI    X020,X084,C040,S040,512
         BTRFLI    X021,X085,C042,S042,512
         BTRFLI    X022,X086,C044,S044,512
         BTRFLI    X023,X087,C046,S046,512
         BTRFLI    X024,X088,C048,S048,512
         BTRFLI    X025,X089,C050,S050,512
         BTRFLI    X026,X090,C052,S052,512
         BTRFLI    X027,X091,C054,S054,512
         BTRFLI    X028,X092,C056,S056,512
         BTRFLI    X029,X093,C058,S058,512
         BTRFLI    X030,X094,C060,S060,512
         BTRFLI    X031,X095,C062,S062,512
         PBY2I     X032,X096,512
         BTRFLI    X033,X097,C066,S066,512
         BTRFLI    X034,X098,C068,S068,512
         BTRFLI    X035,X099,C070,S070,512
         BTRFLI    X036,X100,C072,S072,512
         BTRFLI    X037,X101,C074,S074,512
         BTRFLI    X038,X102,C076,S076,512
         BTRFLI    X039,X103,C078,S078,512
         BTRFLI    X040,X104,C080,S080,512
         BTRFLI    X041,X105,C082,S082,512
         BTRFLI    X042,X106,C084,S084,512
         BTRFLI    X043,X107,C086,S086,512
         BTRFLI    X044,X108,C088,S088,512
         BTRFLI    X045,X109,C090,S090,512
         BTRFLI    X046,X110,C092,S092,512
         BTRFLI    X047,X111,C094,S094,512
         P3BY4I    X048,X112,512
         BTRFLI    X049,X113,C098,S098,512
         BTRFLI    X050,X114,C100,S100,512
         BTRFLI    X051,X115,C102,S102,512
         BTRFLI    X052,X116,C104,S104,512
         BTRFLI    X053,X117,C106,S106,512
```

```
         BTRFLI   X054,X118,C108,S108,512
         BTRFLI   X055,X119,C110,S110,512
         BTRFLI   X056,X120,C112,S112,512
         BTRFLI   X057,X121,C114,S114,512
         BTRFLI   X058,X122,C116,S116,512
         BTRFLI   X059,X123,C118,S118,512
         BTRFLI   X060,X124,C120,S120,512
         BTRFLI   X061,X125,C122,S122,512
         BTRFLI   X062,X126,C124,S124,512
         BTRFLI   X063,X127,C126,S126,512
         ZEROI    X128,X192,512
         BTRFLI   X129,X193,C002,S002,512
         BTRFLI   X130,X194,C004,S004,512
         BTRFLI   X131,X195,C006,S006,512
         BTRFLI   X132,X196,C008,S008,512
         BTRFLI   X133,X197,C010,S010,512
         BTRFLI   X134,X198,C012,S012,512
         BTRFLI   X135,X199,C014,S014,512
         BTRFLI   X136,X200,C016,S016,512
         BTRFLI   X137,X201,C018,S018,512
         BTRFLI   X138,X202,C020,S020,512
         BTRFLI   X139,X203,C022,S022,512
         BTRFLI   X140,X204,C024,S024,512
         BTRFLI   X141,X205,C026,S026,512
         BTRFLI   X142,X206,C028,S028,512
         BTRFLI   X143,X207,C030,S030,512
         PBY4I    X144,X208,512
         BTRFLI   X145,X209,C034,S034,512
         BTRFLI   X146,X210,C036,S036,512
         BTRFLI   X147,X211,C038,S038,512
         BTRFLI   X148,X212,C040,S040,512
         BTRFLI   X149,X213,C042,S042,512
         BTRFLI   X150,X214,C044,S044,512
         BTRFLI   X151,X215,C046,S046,512
         BTRFLI   X152,X216,C048,S048,512
         BTRFLI   X153,X217,C050,S050,512
         BTRFLI   X154,X218,C052,S052,512
         BTRFLI   X155,X219,C054,S054,512
         BTRFLI   X156,X220,C056,S056,512
         BTRFLI   X157,X221,C058,S058,512
         BTRFLI   X158,X222,C060,S060,512
         BTRFLI   X159,X223,C062,S062,512
         PBY2I    X160,X224,512
         BTRFLI   X161,X225,C066,S066,512
         BTRFLI   X162,X226,C068,S068,512
         BTRFLI   X163,X227,C070,S070,512
         BTRFLI   X164,X228,C072,S072,512
         BTRFLI   X165,X229,C074,S074,512
         BTRFLI   X166,X230,C076,S076,512
         BTRFLI   X167,X231,C078,S078,512
         BTRFLI   X168,X232,C080,S080,512
         BTRFLI   X169,X233,C082,S082,512
         BTRFLI   X170,X234,C084,S084,512
         BTRFLI   X171,X235,C086,S086,512
         BTRFLI   X172,X236,C088,S088,512
         BTRFLI   X173,X237,C090,S090,512
         BTRFLI   X174,X238,C092,S092,512
         BTRFLI   X175,X239,C094,S094,512
         P3BY4I   X176,X240,512
         BTRFLI   X177,X241,C098,S098,512
         BTRFLI   X178,X242,C100,S100,512
         BTRFLI   X179,X243,C102,S102,512
         BTRFLI   X180,X244,C104,S104,512
         BTRFLI   X181,X245,C106,S106,512
         BTRFLI   X182,X246,C108,S108,512
         BTRFLI   X183,X247,C110,S110,512
         BTRFLI   X184,X248,C112,S112,512
         BTRFLI   X185,X249,C114,S114,512
         BTRFLI   X186,X250,C116,S116,512
         BTRFLI   X187,X251,C118,S118,512
         BTRFLI   X188,X252,C120,S120,512
         BTRFLI   X189,X253,C122,S122,512
         BTRFLI   X190,X254,C124,S124,512
         BTRFLI   X191,X255,C126,S126,512
*
*****************************************************************
*                                                               *
*        256-POINT FFT KERNEL - STAGE 8                         *
*                                                               *
*****************************************************************
*
         ZEROI    X000,X128,512
         BTRFLI   X001,X129,C001,S001,512
         BTRFLI   X002,X130,C002,S002,512
         BTRFLI   X003,X131,C003,S003,512
         BTRFLI   X004,X132,C004,S004,512
         BTRFLI   X005,X133,C005,S005,512
         BTRFLI   X006,X134,C006,S006,512
         BTRFLI   X007,X135,C007,S007,512
         BTRFLI   X008,X136,C008,S008,512
         BTRFLI   X009,X137,C009,S009,512
         BTRFLI   X010,X138,C010,S010,512
         BTRFLI   X011,X139,C011,S011,512
         BTRFLI   X012,X140,C012,S012,512
         BTRFLI   X013,X141,C013,S013,512
         BTRFLI   X014,X142,C014,S014,512
         BTRFLI   X015,X143,C015,S015,512
         BTRFLI   X016,X144,C016,S016,512
         BTRFLI   X017,X145,C017,S017,512
         BTRFLI   X018,X146,C018,S018,512
         BTRFLI   X019,X147,C019,S019,512
         BTRFLI   X020,X148,C020,S020,512
         BTRFLI   X021,X149,C021,S021,512
         BTRFLI   X022,X150,C022,S022,512
         BTRFLI   X023,X151,C023,S023,512
         BTRFLI   X024,X152,C024,S024,512
         BTRFLI   X025,X153,C025,S025,512
         BTRFLI   X026,X154,C026,S026,512
         BTRFLI   X027,X155,C027,S027,512
         BTRFLI   X028,X156,C028,S028,512
         BTRFLI   X029,X157,C029,S029,512
         BTRFLI   X030,X158,C030,S030,512
         BTRFLI   X031,X159,C031,S031,512
         PBY4I    X032,X160,512
         BTRFLI   X033,X161,C033,S033,512
         BTRFLI   X034,X162,C034,S034,512
         BTRFLI   X035,X163,C035,S035,512
         BTRFLI   X036,X164,C036,S036,512
         BTRFLI   X037,X165,C037,S037,512
         BTRFLI   X038,X166,C038,S038,512
```

```
        BTRFLI  X039,X167,C039,S039,512
        BTRFLI  X040,X168,C040,S040,512
        BTRFLI  X041,X169,C041,S041,512
        BTRFLI  X042,X170,C042,S042,512
        BTRFLI  X043,X171,C043,S043,512
        BTRFLI  X044,X172,C044,S044,512
        BTRFLI  X045,X173,C045,S045,512
        BTRFLI  X046,X174,C046,S046,512
        BTRFLI  X047,X175,C047,S047,512
        BTRFLI  X048,X176,C048,S048,512
        BTRFLI  X049,X177,C049,S049,512
        BTRFLI  X050,X178,C050,S050,512
        BTRFLI  X051,X179,C051,S051,512
        BTRFLI  X052,X180,C052,S052,512
        BTRFLI  X053,X181,C053,S053,512
        BTRFLI  X054,X182,C054,S054,512
        BTRFLI  X055,X183,C055,S055,512
        BTRFLI  X056,X184,C056,S056,512
        BTRFLI  X057,X185,C057,S057,512
        BTRFLI  X058,X186,C058,S058,512
        BTRFLI  X059,X187,C059,S059,512
        BTRFLI  X060,X188,C060,S060,512
        BTRFLI  X061,X189,C061,S061,512
        BTRFLI  X062,X190,C062,S062,512
        BTRFLI  X063,X191,C063,S063,512
        PBY2I   X064,X192,512
        BTRFLI  X065,X193,C065,S065,512
        BTRFLI  X066,X194,C066,S066,512
        BTRFLI  X067,X195,C067,S067,512
        BTRFLI  X068,X196,C068,S068,512
        BTRFLI  X069,X197,C069,S069,512
        BTRFLI  X070,X198,C070,S070,512
        BTRFLI  X071,X199,C071,S071,512
        BTRFLI  X072,X200,C072,S072,512
        BTRFLI  X073,X201,C073,S073,512
        BTRFLI  X074,X202,C074,S074,512
        BTRFLI  X075,X203,C075,S075,512
        BTRFLI  X076,X204,C076,S076,512
        BTRFLI  X077,X205,C077,S077,512
        BTRFLI  X078,X206,C078,S078,512
        BTRFLI  X079,X207,C079,S079,512
        BTRFLI  X080,X208,C080,S080,512
        BTRFLI  X081,X209,C081,S081,512
        BTRFLI  X082,X210,C082,S082,512
        BTRFLI  X083,X211,C083,S083,512
        BTRFLI  X084,X212,C084,S084,512
        BTRFLI  X085,X213,C085,S085,512
        BTRFLI  X086,X214,C086,S086,512
        BTRFLI  X087,X215,C087,S087,512
        BTRFLI  X088,X216,C088,S088,512
        BTRFLI  X089,X217,C089,S089,512
        BTRFLI  X090,X218,C090,S090,512
        BTRFLI  X091,X219,C091,S091,512
        BTRFLI  X092,X220,C092,S092,512
        BTRFLI  X093,X221,C093,S093,512
        BTRFLI  X094,X222,C094,S094,512
        BTRFLI  X095,X223,C095,S095,512
        P3BY4I  X096,X224,512
        BTRFLI  X097,X225,C097,S097,512
        BTRFLI  X098,X226,C098,S098,512
        BTRFLI  X099,X227,C099,S099,512
        BTRFLI  X100,X228,C100,S100,512
        BTRFLI  X101,X229,C101,S101,512
        BTRFLI  X102,X230,C102,S102,512
        BTRFLI  X103,X231,C103,S103,512
        BTRFLI  X104,X232,C104,S104,512
        BTRFLI  X105,X233,C105,S105,512
        BTRFLI  X106,X234,C106,S106,512
        BTRFLI  X107,X235,C107,S107,512
        BTRFLI  X108,X236,C108,S108,512
        BTRFLI  X109,X237,C109,S109,512
        BTRFLI  X110,X238,C110,S110,512
        BTRFLI  X111,X239,C111,S111,512
        BTRFLI  X112,X240,C112,S112,512
        BTRFLI  X113,X241,C113,S113,512
        BTRFLI  X114,X242,C114,S114,512
        BTRFLI  X115,X243,C115,S115,512
        BTRFLI  X116,X244,C116,S116,512
        BTRFLI  X117,X245,C117,S117,512
        BTRFLI  X118,X246,C118,S118,512
        BTRFLI  X119,X247,C119,S119,512
        BTRFLI  X120,X248,C120,S120,512
        BTRFLI  X121,X249,C121,S121,512
        BTRFLI  X122,X250,C122,S122,512
        BTRFLI  X123,X251,C123,S123,512
        BTRFLI  X124,X252,C124,S124,512
        BTRFLI  X125,X253,C125,S125,512
        BTRFLI  X126,X254,C126,S126,512
        BTRFLI  X127,X255,C127,S127,512
        RET
*
*************************************************************
*                                                           *
*    1024-POINT FFT CODE WITH BIT-REVERSED INPUT SAMPLES    *
*                                                           *
*    ALL INPUT REAL AND IMAGINARY DATA POINTS ARE ASSUMED   *
*    TO BE IN CONSECUTIVE LOCATIONS (A TOTAL OF 2048 ) IN   *
*    EXTERNAL DATA MEMORY STARTING FROM LOCATION 1024 IN    *
*    PAGE 8.  OUT OF THE 1024 COMPLEX POINTS, THERE ARE     *
*    ALTOGETHER 496 PAIRS OF INPUT DATA WHICH NEED TO BE    *
*    SCRAMBLED AS SHOWN BELOW.                              *
*                                                           *
*************************************************************
*
FT1024  BITRVI  2,1024,1024
        BITRVI  4,512,1024
        BITRVI  6,1536,1024
        BITRVI  8,256,1024
        BITRVI  10,1280,1024
        BITRVI  12,768,1024
        BITRVI  14,1792,1024
        BITRVI  16,128,1024
        BITRVI  18,1152,1024
        BITRVI  20,640,1024
        BITRVI  22,1664,1024
        BITRVI  24,384,1024
        BITRVI  26,1408,1024
        BITRVI  28,896,1024
        BITRVI  30,1920,1024
        BITRVI  32,64,1024
```

```
BITRVI    34,1088,1024
BITRVI    36,576,1024
BITRVI    38,1600,1024
BITRVI    40,320,1024
BITRVI    42,1344,1024
BITRVI    44,832,1024
BITRVI    46,1856,1024
BITRVI    48,192,1024
BITRVI    50,1216,1024
BITRVI    52,704,1024
BITRVI    54,1728,1024
BITRVI    56,448,1024
BITRVI    58,1472,1024
BITRVI    60,960,1024
BITRVI    62,1984,1024
BITRVI    66,1056,1024
BITRVI    68,544,1024
BITRVI    70,1568,1024
BITRVI    72,288,1024
BITRVI    74,1312,1024
BITRVI    76,800,1024
BITRVI    78,1824,1024
BITRVI    80,160,1024
BITRVI    82,1184,1024
BITRVI    84,672,1024
BITRVI    86,1696,1024
BITRVI    88,416,1024
BITRVI    90,1440,1024
BITRVI    92,928,1024
BITRVI    94,1952,1024
BITRVI    98,1120,1024
BITRVI    100,608,1024
BITRVI    102,1632,1024
BITRVI    104,352,1024
BITRVI    106,1376,1024
BITRVI    108,864,1024
BITRVI    110,1888,1024
BITRVI    112,224,1024
BITRVI    114,1248,1024
BITRVI    116,736,1024
BITRVI    118,1760,1024
BITRVI    120,480,1024
BITRVI    122,1504,1024
BITRVI    124,992,1024
BITRVI    126,2016,1024
BITRVI    130,1040,1024
BITRVI    132,528,1024
BITRVI    134,1552,1024
BITRVI    136,272,1024
BITRVI    138,1296,1024
BITRVI    140,784,1024
BITRVI    142,1808,1024
BITRVI    146,1168,1024
BITRVI    148,656,1024
BITRVI    150,1680,1024
BITRVI    152,400,1024
BITRVI    154,1424,1024
BITRVI    156,912,1024
BITRVI    158,1936,1024
BITRVI    162,1104,1024
BITRVI    164,592,1024
BITRVI    166,1616,1024
BITRVI    168,336,1024
BITRVI    170,1360,1024
BITRVI    172,848,1024
BITRVI    174,1872,1024
BITRVI    176,208,1024
BITRVI    178,1232,1024
BITRVI    180,720,1024
BITRVI    182,1744,1024
BITRVI    184,464,1024
BITRVI    186,1488,1024
BITRVI    188,976,1024
BITRVI    190,2000,1024
BITRVI    194,1072,1024
BITRVI    196,560,1024
BITRVI    198,1584,1024
BITRVI    200,304,1024
BITRVI    202,1328,1024
BITRVI    204,816,1024
BITRVI    206,1840,1024
BITRVI    210,1200,1024
BITRVI    212,688,1024
BITRVI    214,1712,1024
BITRVI    216,432,1024
BITRVI    218,1456,1024
BITRVI    220,944,1024
BITRVI    222,1968,1024
BITRVI    226,1136,1024
BITRVI    228,624,1024
BITRVI    230,1648,1024
BITRVI    232,368,1024
BITRVI    234,1392,1024
BITRVI    236,880,1024
BITRVI    238,1904,1024
BITRVI    242,1264,1024
BITRVI    244,752,1024
BITRVI    246,1776,1024
BITRVI    248,496,1024
BITRVI    250,1520,1024
BITRVI    252,1008,1024
BITRVI    254,2032,1024
BITRVI    258,1032,1024
BITRVI    260,520,1024
BITRVI    262,1544,1024
BITRVI    266,1288,1024
BITRVI    268,776,1024
BITRVI    270,1800,1024
BITRVI    274,1160,1024
BITRVI    276,648,1024
BITRVI    278,1672,1024
BITRVI    280,392,1024
BITRVI    282,1416,1024
BITRVI    284,904,1024
BITRVI    286,1928,1024
BITRVI    290,1096,1024
BITRVI    292,584,1024
BITRVI    294,1608,1024
BITRVI    296,328,1024
BITRVI    298,1352,1024
```

```
BITRVI    300,840,1024
BITRVI    302,1864,1024
BITRVI    306,1224,1024
BITRVI    308,712,1024
BITRVI    310,1736,1024
BITRVI    312,456,1024
BITRVI    314,1480,1024
BITRVI    316,968,1024
BITRVI    318,1992,1024
BITRVI    322,1064,1024
BITRVI    324,552,1024
BITRVI    326,1576,1024
BITRVI    330,1320,1024
BITRVI    332,808,1024
BITRVI    334,1832,1024
BITRVI    338,1192,1024
BITRVI    340,680,1024
BITRVI    342,1704,1024
BITRVI    344,424,1024
BITRVI    346,1448,1024
BITRVI    348,936,1024
BITRVI    350,1960,1024
BITRVI    354,1128,1024
BITRVI    356,616,1024
BITRVI    358,1640,1024
BITRVI    362,1384,1024
BITRVI    364,872,1024
BITRVI    366,1896,1024
BITRVI    370,1256,1024
BITRVI    372,744,1024
BITRVI    374,1768,1024
BITRVI    376,488,1024
BITRVI    378,1512,1024
BITRVI    380,1000,1024
BITRVI    382,2024,1024
BITRVI    386,1048,1024
BITRVI    388,536,1024
BITRVI    390,1560,1024
BITRVI    394,1304,1024
BITRVI    396,792,1024
BITRVI    398,1816,1024
BITRVI    402,1176,1024
BITRVI    404,664,1024
BITRVI    406,1688,1024
BITRVI    410,1432,1024
BITRVI    412,920,1024
BITRVI    414,1944,1024
BITRVI    418,1112,1024
BITRVI    420,600,1024
BITRVI    422,1624,1024
BITRVI    426,1368,1024
BITRVI    428,856,1024
BITRVI    430,1880,1024
BITRVI    434,1240,1024
BITRVI    436,728,1024
BITRVI    438,1752,1024
BITRVI    440,472,1024
BITRVI    442,1496,1024
BITRVI    444,984,1024
BITRVI    446,2008,1024
BITRVI    450,1080,1024
BITRVI    452,568,1024
BITRVI    454,1592,1024
BITRVI    458,1336,1024
BITRVI    460,824,1024
BITRVI    462,1848,1024
BITRVI    466,1208,1024
BITRVI    468,696,1024
BITRVI    470,1720,1024
BITRVI    474,1464,1024
BITRVI    476,952,1024
BITRVI    478,1976,1024
BITRVI    482,1144,1024
BITRVI    484,632,1024
BITRVI    486,1656,1024
BITRVI    490,1400,1024
BITRVI    492,888,1024
BITRVI    494,1912,1024
BITRVI    498,1272,1024
BITRVI    500,760,1024
BITRVI    502,1784,1024
BITRVI    506,1528,1024
BITRVI    508,1016,1024
BITRVI    510,2040,1024
BITRVI    514,1028,1024
BITRVI    518,1540,1024
BITRVI    522,1284,1024
BITRVI    524,772,1024
BITRVI    526,1796,1024
BITRVI    530,1156,1024
BITRVI    532,644,1024
BITRVI    534,1668,1024
BITRVI    538,1412,1024
BITRVI    540,900,1024
BITRVI    542,1924,1024
BITRVI    546,1092,1024
BITRVI    548,580,1024
BITRVI    550,1604,1024
BITRVI    554,1348,1024
BITRVI    556,836,1024
BITRVI    558,1860,1024
BITRVI    562,1220,1024
BITRVI    564,708,1024
BITRVI    566,1732,1024
BITRVI    570,1476,1024
BITRVI    572,964,1024
BITRVI    574,1988,1024
BITRVI    578,1060,1024
BITRVI    582,1572,1024
BITRVI    586,1316,1024
BITRVI    588,804,1024
BITRVI    590,1828,1024
BITRVI    594,1188,1024
BITRVI    596,676,1024
BITRVI    598,1700,1024
BITRVI    602,1444,1024
BITRVI    604,932,1024
BITRVI    606,1956,1024
BITRVI    610,1124,1024
BITRVI    614,1636,1024
```

```
BITRVI    618,1380,1024
BITRVI    620,868,1024
BITRVI    622,1892,1024
BITRVI    626,1252,1024
BITRVI    628,740,1024
BITRVI    630,1764,1024
BITRVI    634,1508,1024
BITRVI    636,996,1024
BITRVI    638,2020,1024
BITRVI    642,1044,1024
BITRVI    646,1556,1024
BITRVI    650,1300,1024
BITRVI    652,788,1024
BITRVI    654,1812,1024
BITRVI    658,1172,1024
BITRVI    662,1684,1024
BITRVI    666,1428,1024
BITRVI    668,916,1024
BITRVI    670,1940,1024
BITRVI    674,1108,1024
BITRVI    678,1620,1024
BITRVI    682,1364,1024
BITRVI    684,852,1024
BITRVI    686,1876,1024
BITRVI    690,1236,1024
BITRVI    692,724,1024
BITRVI    694,1748,1024
BITRVI    698,1492,1024
BITRVI    700,980,1024
BITRVI    702,2004,1024
BITRVI    706,1076,1024
BITRVI    710,1588,1024
BITRVI    714,1332,1024
BITRVI    716,820,1024
BITRVI    718,1844.1024
BITRVI    722,1204,1024
BITRVI    726,1716,1024
BITRVI    730,1460,1024
BITRVI    732,948,1024
BITRVI    734,1972,1024
BITRVI    738,1140,1024
BITRVI    742,1652,1024
BITRVI    746,1396,1024
BITRVI    748,884,1024
BITRVI    750,1908,1024
BITRVI    754,1268,1024
BITRVI    758,1780,1024
BITRVI    762,1524,1024
BITRVI    764,1012,1024
BITRVI    766,2036,1024
BITRVI    770,1036,1024
BITRVI    774,1548,1024
BITRVI    778,1292,1024
BITRVI    782,1804,1024
BITRVI    786,1164,1024
BITRVI    790,1676,1024
BITRVI    794,1420,1024
BITRVI    796,908,1024
BITRVI    798,1932,1024
BITRVI    802,1100,1024
BITRVI    806,1612,1024
BITRVI    810,1356,1024
BITRVI    812,844,1024
BITRVI    814,1868,1024
BITRVI    818,1228,1024
BITRVI    822,1740,1024
BITRVI    826,1484,1024
BITRVI    828,972,1024
BITRVI    830,1996,1024
BITRVI    834,1068,1024
BITRVI    838,1580,1024
BITRVI    842,1324,1024
BITRVI    846,1836,1024
BITRVI    850,1196,1024
BITRVI    854,1708,1024
BITRVI    858,1452,1024
BITRVI    860,940,1024
BITRVI    862,1964,1024
BITRVI    866,1132,1024
BITRVI    870,1644,1024
BITRVI    874,1388,1024
BITRVI    878,1900,1024
BITRVI    882,1260,1024
BITRVI    886,1772,1024
BITRVI    890,1516,1024
BITRVI    892,1004,1024
BITRVI    894,2028,1024
BITRVI    898,1052,1024
BITRVI    902,1564,1024
BITRVI    906,1308,1024
BITRVI    910,1820,1024
BITRVI    914,1180,1024
BITRVI    918,1692,1024
BITRVI    922,1436,1024
BITRVI    926,1948,1024
BITRVI    930,1116,1024
BITRVI    934,1628,1024
BITRVI    938,1372,1024
BITRVI    942,1884,1024
BITRVI    946,1244,1024
BITRVI    950,1756,1024
BITRVI    954,1500,1024
BITRVI    956,988,1024
BITRVI    958,2012,1024
BITRVI    962,1084,1024
BITRVI    966,1596,1024
BITRVI    970,1340,1024
BITRVI    974,1852,1024
BITRVI    978,1212,1024
BITRVI    982,1724,1024
BITRVI    986,1468,1024
BITRVI    990,1980,1024
BITRVI    994,1148,1024
BITRVI    998,1660,1024
BITRVI    1002,1404,1024
BITRVI    1006,1916,1024
BITRVI    1010,1276,1024
BITRVI    1014,1788,1024
BITRVI    1018,1532,1024
BITRVI    1022,2044,1024
```

```
BITRVI	1030,1538,1024
BITRVI	1034,1282,1024
BITRVI	1038,1794,1024
BITRVI	1042,1154,1024
BITRVI	1046,1666,1024
BITRVI	1050,1410,1024
BITRVI	1054,1922,1024
BITRVI	1058,1090,1024
BITRVI	1062,1602,1024
BITRVI	1066,1346,1024
BITRVI	1070,1858,1024
BITRVI	1074,1218,1024
BITRVI	1078,1730,1024
BITRVI	1082,1474,1024
BITRVI	1086,1986,1024
BITRVI	1094,1570,1024
BITRVI	1098,1314,1024
BITRVI	1102,1826,1024
BITRVI	1106,1186,1024
BITRVI	1110,1698,1024
BITRVI	1114,1442,1024
BITRVI	1118,1954,1024
BITRVI	1126,1634,1024
BITRVI	1130,1378,1024
BITRVI	1134,1890,1024
BITRVI	1138,1250,1024
BITRVI	1142,1762,1024
BITRVI	1146,1506,1024
BITRVI	1150,2018,1024
BITRVI	1158,1554,1024
BITRVI	1162,1298,1024
BITRVI	1166,1810,1024
BITRVI	1174,1682,1024
BITRVI	1178,1426,1024
BITRVI	1182,1938,1024
BITRVI	1190,1618,1024
BITRVI	1194,1362,1024
BITRVI	1198,1874,1024
BITRVI	1202,1234,1024
BITRVI	1206,1746,1024
BITRVI	1210,1490,1024
BITRVI	1214,2002,1024
BITRVI	1222,1586,1024
BITRVI	1226,1330,1024
BITRVI	1230,1842,1024
BITRVI	1238,1714,1024
BITRVI	1242,1458,1024
BITRVI	1246,1970,1024
BITRVI	1254,1650,1024
BITRVI	1258,1394,1024
BITRVI	1262,1906,1024
BITRVI	1270,1778,1024
BITRVI	1274,1522,1024
BITRVI	1278,2034,1024
BITRVI	1286,1546,1024
BITRVI	1294,1802,1024
BITRVI	1302,1674,1024
BITRVI	1306,1418,1024
BITRVI	1310,1930,1024
BITRVI	1318,1610,1024
BITRVI	1322,1354,1024
BITRVI	1326,1866,1024
BITRVI	1334,1738,1024
BITRVI	1338,1482,1024
BITRVI	1342,1994,1024
BITRVI	1350,1578,1024
BITRVI	1358,1834,1024
BITRVI	1366,1706,1024
BITRVI	1370,1450,1024
BITRVI	1374,1962,1024
BITRVI	1382,1642,1024
BITRVI	1390,1898,1024
BITRVI	1398,1770,1024
BITRVI	1402,1514,1024
BITRVI	1406,2026,1024
BITRVI	1414,1562,1024
BITRVI	1422,1818,1024
BITRVI	1430,1690,1024
BITRVI	1438,1946,1024
BITRVI	1446,1626,1024
BITRVI	1454,1882,1024
BITRVI	1462,1754,1024
BITRVI	1466,1498,1024
BITRVI	1470,2010,1024
BITRVI	1478,1594,1024
BITRVI	1486,1850,1024
BITRVI	1494,1722,1024
BITRVI	1502,1978,1024
BITRVI	1510,1658,1024
BITRVI	1518,1914,1024
BITRVI	1526,1786,1024
BITRVI	1534,2042,1024
BITRVI	1550,1798,1024
BITRVI	1558,1670,1024
BITRVI	1566,1926,1024
BITRVI	1574,1606,1024
BITRVI	1582,1862,1024
BITRVI	1590,1734,1024
BITRVI	1598,1990,1024
BITRVI	1614,1830,1024
BITRVI	1622,1702,1024
BITRVI	1630,1958,1024
BITRVI	1646,1894,1024
BITRVI	1654,1766,1024
BITRVI	1662,2022,1024
BITRVI	1678,1814,1024
BITRVI	1694,1942,1024
BITRVI	1710,1878,1024
BITRVI	1718,1750,1024
BITRVI	1726,2006,1024
BITRVI	1742,1846,1024
BITRVI	1758,1974,1024
BITRVI	1774,1910,1024
BITRVI	1790,2038,1024
BITRVI	1822,1934,1024
BITRVI	1838,1870,1024
BITRVI	1854,1998,1024
BITRVI	1886,1966,1024
BITRVI	1918,2030,1024
BITRVI	1982,2014,1024
```

```
*
****************************************************************
*                                                              *
*    THE FIRST 8 STAGES OF THE 10-STAGE 1024-POINT COMPLEX     *
*    FFT WILL BE PERFORMED AS 4 SEPARATE 256-POINT COMPLEX     *
*    FFT'S USING THE ON-CHIP DATA BLOCKS B0 AND B1.  THE       *
*    KERNEL CODE BELOW WILL THEREFORE BE CALLED 4 TIMES TO     *
*    ACCOMPLISH THIS WHILE EXTERNAL DATA MEMORY WILL HAVE      *
*    MOVE ON-CHIP USING THE BLKD INSTRUCTION.                  *
*                                                              *
****************************************************************
*
        LARP    AR1             BLOCK MOVE FIRST GROUP OF 256
        LRLK    AR1,512         COMPLEX POINTS (1024-1535) FROM
        RPTK    255             EXTERNAL RAM TO ON-CHIP RAM
        BLKD    1024,*+
        RPTK    255
        BLKD    1280,*+
        CALL    KNL256          EXECUTE 256-POINT KERNEL FFT
        LRLK    AR1,1024        AND RETURN RESULTS TO
        RPTK    255             EXTERNAL RAM (1024-1535)
        BLKD    512,*+
        RPTK    255
        BLKD    768,*+
*
        LRLK    AR1,512         BLOCK MOVE SECOND GROUP OF 256
        RPTK    255             COMPLEX POINTS (1536-2047) FROM
        BLKD    1536,*+         EXTERNAL RAM TO ON-CHIP RAM
        RPTK    255
        BLKD    1792,*+
        CALL    KNL256          EXECUTE 256-POINT KERNEL FFT
        LRLK    AR1,1536        AND RETURN RESULTS TO
        RPTK    255             EXTERNAL RAM (1536-2047)
        BLKD    512,*+
        RPTK    255
        BLKD    768,*+
*
        LRLK    AR1,512         BLOCK MOVE THIRD GROUP OF 256
        RPTK    255             COMPLEX POINTS (2048-2559) FROM
        BLKD    2048,*+         EXTERNAL RAM TO ON-CHIP RAM
        RPTK    255
        BLKD    2304,*+
        CALL    KNL256          EXECUTE 256-POINT KERNEL FFT
        LRLK    AR1,2048        AND RETURN RESULTS TO
        RPTK    255             EXTERNAL RAM (2048-2559)
        BLKD    512,*+
        RPTK    255
        BLKD    768,*+
*
        LRLK    AR1,512         BLOCK MOVE FOURTH GROUP OF 256
        RPTK    255             COMPLEX POINTS (2560-3071) FROM
        BLKD    2560,*+         EXTERNAL RAM TO ON-CHIP RAM
        RPTK    255
        BLKD    2816,*+
        CALL    KNL256          EXECUTE 256-POINT KERNEL FFT
        LRLK    AR1,2560        AND RETURN RESULTS TO
        RPTK    255             EXTERNAL RAM (2560-3071)
        BLKD    512,*+
        RPTK    255
        BLKD    768,*+
```

```
*
****************************************************************
*                                                              *
*    PERFORM STAGE 9 OF THE 1024-POINT FFT -- TWIDDLE          *
*    FACTOR VALUES ARE TABLE-READ FROM EXTERNAL PROGRAM        *
*    MEMORY TO ON-CHIP RAM AND THE GENERAL 'BUTTERFLY'         *
*    SUBROUTINE IS EXECUTED 512 TIMES WITH ALL DATA IN         *
*    EXTERNAL DATA MEMORY.  A SPEED IMPROVEMENT CAN BE         *
*    ACHIEVED BY MOVING FFT DATA ON-CHIP INSTEAD OF TWIDDLE    *
*    FACTOR VALUES FOR EXECUTION.  HOWEVER, THE TWIDDLE        *
*    FACTOR VALUES WILL HAVE TO BE STORED IN EXTERNAL DATA     *
*    RAM.                                                      *
*                                                              *
****************************************************************
*
STAGE9  LRLK    AR0,512         INITIALISE TWIDDLE FACTORS
        LRLK    AR1,255         SET UP TWIDDLE TABLE SIZE
        LARP    AR0             USE AR0 TO POINT AT TABLE
        LALK    W000            SET UP TWIDDLE TABLE ADDRESS
LOOP    TBLR    *+              AND STORE IN INTERNAL RAM
        ADLK    1
        TBLR    *+,AR1          AND STORE IN INTERNAL RAM
        ADLK    3
        BANZ    LOOP,*-,AR0
*
        LRLK    AR0,512         INITIALISE STEP SIZE
        LRLK    AR1,512         AR1 POINTS AT TWIDDLE FACTORS
        LRLK    AR2,1024+512    AR2 POINTS AT REAL DATA
        LRLK    AR3,255         INITIALISE LOOP COUNT
        LARP    AR2
LOOP1   CALL    BTRFLY          PERFORM LOOPED FFT FOR STAGE 9
        LARP    AR3
        BANZ    LOOP1,*-,AR2
*
        LRLK    AR1,512         AR1 POINTS AT TWIDDLE FACTORS
        LRLK    AR2,2048+512    AR2 POINTS AT REAL DATA
        LRLK    AR3,255         INITIALISE LOOP COUNT
        LARP    AR2
LOOP2   CALL    BTRFLY          PERFORM LOOPED FFT FOR STAGE 9
        LARP    AR3
        BANZ    LOOP2,*-,AR2
*
****************************************************************
*                                                              *
*    PERFORM STAGE 10 OF THE 1024-POINT FFT -- TWIDDLE         *
*    FACTOR VALUES ARE TABLE-READ FROM EXTERNAL PROGRAM        *
*    MEMORY TO ON-CHIP RAM AND THE GENERAL 'BUTTERFLY'         *
*    SUBROUTINE IS EXECUTED 512 TIMES WITH ALL DATA IN         *
*    EXTERNAL DATA MEMORY.  A SPEED IMPROVEMENT CAN BE         *
*    ACHIEVED BY MOVING FFT DATA ON-CHIP INSTEAD OF TWIDDLE    *
*    FACTOR VALUES FOR EXECUTION.  HOWEVER, THE TWIDDLE        *
*    FACTOR VALUES WILL HAVE TO BE STORED IN EXTERNAL DATA     *
*    RAM.                                                      *
*                                                              *
****************************************************************
*
STGE10  LARP  . AR0
        LRLK    AR0,512         INITIALISE TWIDDLE FACTORS
        RPTK    255             AND STORE IN INTERNAL RAM.
        BLKP    W000,*+
```

```
         RPTK     255
         BLKP     W128,*+
*
         LARP     AR0
         LRLK     AR0,1024       INITIALISE STEP SIZE
         LRLK     AR1,512        AR1 POINTS AT TWIDDLE FACTORS
         LRLK     AR2,2048       AR2 POINTS AT REAL DATA
         LRLK     AR3,255        INITIALISE LOOP COUNT
         LARP     AR2
LOOP3    CALL     BTRFLY         PERFORM LOOPED FFT FOR STAGE 10
         LARP     AR3
         BANZ     LOOP3,*-,AR2
*
         LARP     AR0
         LRLK     AR0,512        INITIALISE TWIDDLE FACTORS
         RPTK     255            AND STORE IN INTERNAL RAM.
         BLKP     W256,*+
         RPTK     255
         BLKP     W384,*+
*
         LARP     AR0
         LRLK     AR0,1024       INITIALISE STEP SIZE
         LRLK     AR1,512        AR1 POINTS AT TWIDDLE FACTORS
         LRLK     AR2,2048+512   AR2 POINTS AT REAL DATA
         LRLK     AR3,255        INITIALISE LOOP COUNT
         LARP     AR2
LOOP4    CALL     BTRFLY         PERFORM LOOPED FFT FOR STAGE 10
         LARP     AR3
         BANZ     LOOP4,*-,AR2
         B        DONE
*
*************************************************************
*                                                           *
*        RADIX-2 GENERAL BUTTERFLY SUBROUTINE               *
*                                                           *
*                                                           *
*       P=(PR+jQI)   P+Q*W=(PR+Re[Q*W])+j(PI+Im[Q*W])       *
*                 \ /                                       *
*                 / \                                       *
*       Q=(QR+jQI)   P-Q*W=(PR-Re[Q*W])+j(PI+Im[Q*W])       *
*                                                           *
*           -j(2(pi)/N)k                                       *
*       W =e          =COS(X)-jSIN(X)                       *
*        N                                                  *
*                     =WR+jWI                               *
*                                                           *
*************************************************************
*
*       CALCULATE QR*COS(X) + QI*SIN(X) AND STORE RESULT IN QR
*
BTRFLY  LT   *+,AR1      LOAD T-REGISTER WITH QR
        MPY  *+,AR2      P-REGISTER = (1/2)(QR*COSX)
        LTP  *-,AR1      ACC= (1/2)(QR*COSX); LOAD TR WITH QI
        MPY  *-          P-REGISTER = (1/2)(QI*SINX)
        APAC             ACC= (1/2)(QR*COSX+QI*SINX)
        MPY  *+,AR2      P-REGISTER = (1/2)(QI*COSX)
        LT   *           LOAD T-REGISTER WITH QR
        SACH *+,0,AR1    QR = (1/2)(QR*COSX+QI*SINX)
*
*       CALCULATE QI*COS(X) - QR*SIN(X) AND STORE RESULT IN QI
*
        PAC              ACC= (1/2)(QI*COSX)
        MPY  *+,AR2      P-REGISTER = (1/2)(QR*SINX)
        SPAC             ACC= (1/2)(QI*COSX-QR*SINX)
        SACH *0-         QI = (1/2)(QI*COSX-QR*SINX)
*
*       CALCULATE Re[Pm+1] & Re[Qm+1] STORE RESULTS in PR & QR
*
        MAR  *-
        LAC  *0+,14      ACC= (1/4)PR
        ADD  *0-,15      ACC= (1/4)[PR+(QR*COSX+QI*SINX)]
        SACH *0+,1       PR = (1/2)[PR+(QR*COSX+QI*SINX)]
        SUBH *           ACC= (1/4)[PR-(QR*COSX+QI*SINX)]
        SACH *0-,1       QR = (1/2)[PR-(QR*COSX+QI*SINX)]
*
*       CALCULATE Im[Pm+1] & Im[Qm+1] STORE RESULTS in PI & QI
*
        MAR  *+
        LAC  *0+,14      ACC= (1/4)PI
        ADD  *0-,15      ACC= (1/4)[PI+(QI*COSX-QR*SINX)]
        SACH *0+,1       PI = (1/2)[PI+(QI*COSX-QR*SINX)]
        SUBH *           ACC= (1/4)[PI-(QI*COSX-QR*SINX)]
        SACH *+,1        QI = (1/2)[PI-(QI*COSX-QR*SINX)]
        RET
*
*************************************************************
*                                                           *
*  16-BIT TWIDDLE FACTORS FOR 1024-POINT COMPLEX FFT        *
*                                                           *
*************************************************************
*
W000     DATA     32767,0
W001     DATA     32766,201
W002     DATA     32765,402
W003     DATA     32762,603
W004     DATA     32758,804
W005     DATA     32752,1005
W006     DATA     32746,1206
W007     DATA     32738,1407
W008     DATA     32728,1608
W009     DATA     32718,1809
W010     DATA     32706,2009
W011     DATA     32693,2210
W012     DATA     32679,2410
W013     DATA     32664,2611
W014     DATA     32647,2811
W015     DATA     32629,3012
W016     DATA     32610,3212
W017     DATA     32590,3412
W018     DATA     32568,3612
W019     DATA     32545,3811
W020     DATA     32521,4011
W021     DATA     32496,4210
W022     DATA     32470,4410
W023     DATA     32442,4609
W024     DATA     32413,4808
W025     DATA     32383,5007
W026     DATA     32352,5205
W027     DATA     32319,5404
W028     DATA     32285,5602
```

```
W029    DATA    32250,5800
W030    DATA    32214,5998
W031    DATA    32177,6195
W032    DATA    32138,6393
W033    DATA    32098,6590
W034    DATA    32057,6786
W035    DATA    32015,6983
W036    DATA    31972,7179
W037    DATA    31927,7375
W038    DATA    31881,7571
W039    DATA    31834,7767
W040    DATA    31786,7962
W041    DATA    31736,8157
W042    DATA    31686,8351
W043    DATA    31634,8546
W044    DATA    31581,8739
W045    DATA    31527,8933
W046    DATA    31471,9126
W047    DATA    31415,9319
W048    DATA    31357,9512
W049    DATA    31298,9704
W050    DATA    31238,9896
W051    DATA    31176,10088
W052    DATA    31114,10279
W053    DATA    31050,10469
W054    DATA    30985,10660
W055    DATA    30919,10850
W056    DATA    30852,11039
W057    DATA    30784,11228
W058    DATA    30715,11417
W059    DATA    30644,11605
W060    DATA    30572,11793
W061    DATA    30499,11980
W062    DATA    30425,12167
W063    DATA    30350,12354
W064    DATA    30274,12540
W065    DATA    30196,12725
W066    DATA    30117,12910
W067    DATA    30038,13095
W068    DATA    29957,13279
W069    DATA    29875,13462
W070    DATA    29791,13645
W071    DATA    29707,13828
W072    DATA    29622,14010
W073    DATA    29535,14191
W074    DATA    29448,14372
W075    DATA    29359,14553
W076    DATA    29269,14733
W077    DATA    29178,14912
W078    DATA    29086,15091
W079    DATA    28993,15269
W080    DATA    28899,15447
W081    DATA    28803,15624
W082    DATA    28707,15800
W083    DATA    28609,15976
W084    DATA    28511,16151
W085    DATA    28411,16326
W086    DATA    28310,16500
W087    DATA    28209,16673
W088    DATA    28106,16846
W089    DATA    28002,17018
W090    DATA    27897,17190
W091    DATA    27791,17360
W092    DATA    27684,17531
W093    DATA    27576,17700
W094    DATA    27467,17869
W095    DATA    27357,18037
W096    DATA    27245,18205
W097    DATA    27133,18372
W098    DATA    27020,18538
W099    DATA    26906,18703
W100    DATA    26790,18868
W101    DATA    26674,19032
W102    DATA    26557,19195
W103    DATA    26439,19358
W104    DATA    26319,19520
W105    DATA    26199,19681
W106    DATA    26078,19841
W107    DATA    25956,20001
W108    DATA    25832,20160
W109    DATA    25708,20318
W110    DATA    25583,20475
W111    DATA    25457,20632
W112    DATA    25330,20788
W113    DATA    25202,20943
W114    DATA    25073,21097
W115    DATA    24943,21250
W116    DATA    24812,21403
W117    DATA    24680,21555
W118    DATA    24548,21706
W119    DATA    24414,21856
W120    DATA    24279,22005
W121    DATA    24144,22154
W122    DATA    24007,22302
W123    DATA    23870,22449
W124    DATA    23732,22595
W125    DATA    23593,22740
W126    DATA    23453,22884
W127    DATA    23312,23028
W128    DATA    23170,23170
W129    DATA    23028,23312
W130    DATA    22884,23453
W131    DATA    22740,23593
W132    DATA    22595,23732
W133    DATA    22449,23870
W134    DATA    22302,24007
W135    DATA    22154,24144
W136    DATA    22005,24279
W137    DATA    21856,24414
W138    DATA    21706,24548
W139    DATA    21555,24680
W140    DATA    21403,24812
W141    DATA    21250,24943
W142    DATA    21097,25073
W143    DATA    20943,25202
W144    DATA    20788,25330
W145    DATA    20632,25457
W146    DATA    20475,25583
W147    DATA    20318,25708
W148    DATA    20160,25832
```

```
W149    DATA    20001,25956
W150    DATA    19841,26078
W151    DATA    19681,26199
W152    DATA    19520,26319
W153    DATA    19358,26439
W154    DATA    19195,26557
W155    DATA    19032,26674
W156    DATA    18868,26790
W157    DATA    18703,26906
W158    DATA    18538,27020
W159    DATA    18372,27133
W160    DATA    18205,27245
W161    DATA    18037,27357
W162    DATA    17869,27467
W163    DATA    17700,27576
W164    DATA    17531,27684
W165    DATA    17360,27791
W166    DATA    17190,27897
W167    DATA    17018,28002
W168    DATA    16846,28106
W169    DATA    16673,28209
W170    DATA    16500,28311
W171    DATA    16326,28411
W172    DATA    16151,28511
W173    DATA    15976,28609
W174    DATA    15800,28707
W175    DATA    15624,28803
W176    DATA    15447,28899
W177    DATA    15269,28993
W178    DATA    15091,29086
W179    DATA    14912,29178
W180    DATA    14733,29269
W181    DATA    14553,29359
W182    DATA    14372,29448
W183    DATA    14191,29535
W184    DATA    14010,29622
W185    DATA    13828,29707
W186    DATA    13645,29791
W187    DATA    13462,29875
W188    DATA    13279,29957
W189    DATA    13095,30038
W190    DATA    12910,30117
W191    DATA    12725,30196
W192    DATA    12540,30274
W193    DATA    12354,30350
W194    DATA    12167,30425
W195    DATA    11980,30499
W196    DATA    11793,30572
W197    DATA    11605,30644
W198    DATA    11417,30715
W199    DATA    11228,30784
W200    DATA    11039,30852
W201    DATA    10850,30920
W202    DATA    10660,30986
W203    DATA    10469,31050
W204    DATA    10279,31114
W205    DATA    10087,31176
W206    DATA    9896,31238
W207    DATA    9704,31298
W208    DATA    9512,31357
W209    DATA    9319,31415
W210    DATA    9126,31471
W211    DATA    8933,31527
W212    DATA    8739,31581
W213    DATA    8546,31634
W214    DATA    8351,31686
W215    DATA    8157,31736
W216    DATA    7962,31786
W217    DATA    7767,31834
W218    DATA    7571,31881
W219    DATA    7375,31927
W220    DATA    7179,31972
W221    DATA    6983,32015
W222    DATA    6786,32057
W223    DATA    6590,32098
W224    DATA    6393,32138
W225    DATA    6195,32177
W226    DATA    5998,32214
W227    DATA    5800,32250
W228    DATA    5602,32285
W229    DATA    5404,32319
W230    DATA    5205,32352
W231    DATA    5007,32383
W232    DATA    4808,32413
W233    DATA    4609,32442
W234    DATA    4410,32470
W235    DATA    4211,32496
W236    DATA    4011,32521
W237    DATA    3811,32545
W238    DATA    3612,32568
W239    DATA    3412,32590
W240    DATA    3212,32610
W241    DATA    3012,32629
W242    DATA    2811,32647
W243    DATA    2611,32664
W244    DATA    2410,32679
W245    DATA    2210,32693
W246    DATA    2009,32706
W247    DATA    1809,32718
W248    DATA    1608,32728
W249    DATA    1407,32738
W250    DATA    1206,32746
W251    DATA    1005,32752
W252    DATA    804,32758
W253    DATA    603,32762
W254    DATA    402,32765
W255    DATA    201,32766
W256    DATA    0,32767
W257    DATA    -201,32766
W258    DATA    -402,32765
W259    DATA    -603,32762
W260    DATA    -804,32758
W261    DATA    -1005,32752
W262    DATA    -1206,32746
W263    DATA    -1407,32738
W264    DATA    -1608,32728
W265    DATA    -1809,32718
W266    DATA    -2009,32706
W267    DATA    -2210,32693
W268    DATA    -2410,32679
```

```
W269    DATA    -2611,32663
W270    DATA    -2811,32647
W271    DATA    -3012,32629
W272    DATA    -3212,32610
W273    DATA    -3412,32590
W274    DATA    -3612,32568
W275    DATA    -3811,32545
W276    DATA    -4011,32521
W277    DATA    -4210,32496
W278    DATA    -4410,32470
W279    DATA    -4609,32442
W280    DATA    -4808,32413
W281    DATA    -5007,32383
W282    DATA    -5205,32352
W283    DATA    -5404,32319
W284    DATA    -5602,32285
W285    DATA    -5800,32250
W286    DATA    -5998,32214
W287    DATA    -6195,32177
W288    DATA    -6393,32138
W289    DATA    -6590,32098
W290    DATA    -6786,32057
W291    DATA    -6983,32015
W292    DATA    -7179,31972
W293    DATA    -7375,31927
W294    DATA    -7571,31881
W295    DATA    -7767,31834
W296    DATA    -7962,31786
W297    DATA    -8157,31736
W298    DATA    -8351,31686
W299    DATA    -8546,31634
W300    DATA    -8739,31581
W301    DATA    -8933,31527
W302    DATA    -9126,31471
W303    DATA    -9319,31415
W304    DATA    -9512,31357
W305    DATA    -9704,31298
W306    DATA    -9896,31238
W307    DATA    -10087,31176
W308    DATA    -10279,31114
W309    DATA    -10469,31050
W310    DATA    -10660,30986
W311    DATA    -10850,30920
W312    DATA    -11039,30852
W313    DATA    -11228,30784
W314    DATA    -11417,30715
W315    DATA    -11605,30644
W316    DATA    -11793,30572
W317    DATA    -11980,30499
W318    DATA    -12167,30425
W319    DATA    -12354,30350
W320    DATA    -12540,30274
W321    DATA    -12725,30196
W322    DATA    -12910,30117
W323    DATA    -13095,30037
W324    DATA    -13279,29957
W325    DATA    -13462,29875
W326    DATA    -13645,29791
W327    DATA    -13828,29707
W328    DATA    -14010,29622
W329    DATA    -14191,29535
W330    DATA    -14372,29448
W331    DATA    -14553,29359
W332    DATA    -14733,29269
W333    DATA    -14912,29178
W334    DATA    -15091,29086
W335    DATA    -15269,28993
W336    DATA    -15447,28899
W337    DATA    -15624,28803
W338    DATA    -15800,28707
W339    DATA    -15976,28609
W340    DATA    -16151,28511
W341    DATA    -16326,28411
W342    DATA    -16500,28310
W343    DATA    -16673,28209
W344    DATA    -16846,28106
W345    DATA    -17018,28002
W346    DATA    -17190,27897
W347    DATA    -17360,27791
W348    DATA    -17531,27684
W349    DATA    -17700,27576
W350    DATA    -17869,27467
W351    DATA    -18037,27357
W352    DATA    -18205,27245
W353    DATA    -18372,27133
W354    DATA    -18538,27020
W355    DATA    -18703,26906
W356    DATA    -18868,26790
W357    DATA    -19032,26674
W358    DATA    -19195,26557
W359    DATA    -19358,26439
W360    DATA    -19520,26319
W361    DATA    -19681,26199
W362    DATA    -19841,26078
W363    DATA    -20001,25956
W364    DATA    -20160,25832
W365    DATA    -20318,25708
W366    DATA    -20475,25583
W367    DATA    -20632,25457
W368    DATA    -20788,25330
W369    DATA    -20943,25202
W370    DATA    -21097,25073
W371    DATA    -21250,24943
W372    DATA    -21403,24812
W373    DATA    -21555,24680
W374    DATA    -21706,24548
W375    DATA    -21856,24414
W376    DATA    -22005,24279
W377    DATA    -22154,24144
W378    DATA    -22302,24007
W379    DATA    -22449,23870
W380    DATA    -22595,23732
W381    DATA    -22740,23593
W382    DATA    -22884,23453
W383    DATA    -23028,23312
W384    DATA    -23170,23170
W385    DATA    -23312,23028
W386    DATA    -23453,22884
W387    DATA    -23593,22740
W388    DATA    -23732,22595
```

```
W389    DATA    -23870,22449
W390    DATA    -24007,22302
W391    DATA    -24144,22154
W392    DATA    -24279,22005
W393    DATA    -24414,21856
W394    DATA    -24548,21706
W395    DATA    -24680,21555
W396    DATA    -24812,21403
W397    DATA    -24943,21250
W398    DATA    -25073,21097
W399    DATA    -25202,20943
W400    DATA    -25330,20788
W401    DATA    -25457,20632
W402    DATA    -25583,20475
W403    DATA    -25708,20318
W404    DATA    -25832,20160
W405    DATA    -25956,20001
W406    DATA    -26078,19841
W407    DATA    -26199,19681
W408    DATA    -26319,19520
W409    DATA    -26439,19358
W410    DATA    -26557,19195
W411    DATA    -26674,19032
W412    DATA    -26790,18868
W413    DATA    -26906,18703
W414    DATA    -27020,18538
W415    DATA    -27133,18372
W416    DATA    -27245,18205
W417    DATA    -27357,18037
W418    DATA    -27467,17869
W419    DATA    -27576,17700
W420    DATA    -27684,17530
W421    DATA    -27791,17360
W422    DATA    -27897,17190
W423    DATA    -28002,17018
W424    DATA    -28106,16846
W425    DATA    -28209,16673
W426    DATA    -28311,16500
W427    DATA    -28411,16326
W428    DATA    -28511,16151
W429    DATA    -28609,15976
W430    DATA    -28707,15800
W431    DATA    -28803,15624
W432    DATA    -28899,15447
W433    DATA    -28993,15269
W434    DATA    -29086,15091
W435    DATA    -29178,14912
W436    DATA    -29269,14733
W437    DATA    -29359,14553
W438    DATA    -29448,14372
W439    DATA    -29535,14191
W440    DATA    -29622,14010
W441    DATA    -29707,13828
W442    DATA    -29791,13645
W443    DATA    -29875,13462
W444    DATA    -29957,13279
W445    DATA    -30038,13095
W446    DATA    -30117,12910
W447    DATA    -30196,12725
W448    DATA    -30274,12540
W449    DATA    -30350,12354
W450    DATA    -30425,12167
W451    DATA    -30499,11980
W452    DATA    -30572,11793
W453    DATA    -30644,11605
W454    DATA    -30715,11417
W455    DATA    -30784,11228
W456    DATA    -30852,11039
W457    DATA    -30920,10850
W458    DATA    -30985,10660
W459    DATA    -31050,10469
W460    DATA    -31114,10279
W461    DATA    -31176,10087
W462    DATA    -31238,9896
W463    DATA    -31298,9704
W464    DATA    -31357,9512
W465    DATA    -31415,9319
W466    DATA    -31471,9126
W467    DATA    -31527,8933
W468    DATA    -31581,8739
W469    DATA    -31634,8546
W470    DATA    -31686,8351
W471    DATA    -31736,8157
W472    DATA    -31786,7962
W473    DATA    -31834,7767
W474    DATA    -31881,7571
W475    DATA    -31927,7375
W476    DATA    -31972,7179
W477    DATA    -32015,6983
W478    DATA    -32057,6786
W479    DATA    -32098,6589
W480    DATA    -32138,6393
W481    DATA    -32177,6195
W482    DATA    -32214,5997
W483    DATA    -32250,5800
W484    DATA    -32285,5602
W485    DATA    -32319,5404
W486    DATA    -32352,5205
W487    DATA    -32383,5007
W488    DATA    -32413,4808
W489    DATA    -32442,4609
W490    DATA    -32470,4410
W491    DATA    -32496,4210
W492    DATA    -32521,4011
W493    DATA    -32545,3811
W494    DATA    -32568,3612
W495    DATA    -32590,3412
W496    DATA    -32610,3212
W497    DATA    -32629,3012
W498    DATA    -32647,2811
W499    DATA    -32664,2611
W500    DATA    -32679,2410
W501    DATA    -32693,2210
W502    DATA    -32706,2009
W503    DATA    -32718,1809
W504    DATA    -32728,1608
W505    DATA    -32738,1407
W506    DATA    -32746,1206
W507    DATA    -32752,1005
W508    DATA    -32758,804
W509    DATA    -32762,603
W510    DATA    -32765,402
W511    DATA    -32766,201
DONE    NOP
        END
```

APPENDIX E
A 256-POINT, RADIX-4 DIF FFT IMPLEMENTATION

```
                        A P P E N D I X   E
                        -------------------

         IDT 'S4FFT256'
*
*
*    COOLEY-TUKEY RADIX-4, DIF FFT PROGRAM - 256-POINT STRAIGHT-LINE.
*
*       FOUR STAGES OF THREE TYPES RADIX-4 BUTTERFLIES -
*       ZERO, SPECIAL, AND NORMAL - IMPLEMENTED WITH MACROS.
*       COMPLEX INPUT DATA LOCATED ON PAGES 4-7  OF DATA MEMORY.
*       RESULTS ARE LEFT IN DATA RAM.
*       USES 13-BIT COEFFICIENTS FROM MPYK INSTRUCTIONS.
*       INTERMEDIATE VALUES ARE SCALED BY 1/4 AT EACH STAGE TO PREVENT
*        OVERFLOW.
*
********************************************************************************
*
* DATA MEMORY ALLOCATION.
*
********************************************************************************
N      EQU  256        * FFT LENGTH
*
T1     EQU  96         * TEMPORARY LOCATIONS ADDRESSED
T2     EQU  97         * BY AUXILIARY REGISTERS.
*
X0      EQU  512
X1      EQU  514
X2      EQU  516
X3      EQU  518
X4      EQU  520
X5      EQU  522
X6      EQU  524
X7      EQU  526
X8      EQU  528
X9      EQU  530
X10     EQU  532
X11     EQU  534
X12     EQU  536
X13     EQU  538
X14     EQU  540
X15     EQU  542
X16     EQU  544
X17     EQU  546
X18     EQU  548
X19     EQU  550
X20     EQU  552
X21     EQU  554
X22     EQU  556
X23     EQU  558
X24     EQU  560
X25     EQU  562
X26     EQU  564
X27     EQU  566
X28     EQU  568
X29     EQU  570
```

```
X30     EQU  572
X31     EQU  574
X32     EQU  576
X33     EQU  578
X34     EQU  580
X35     EQU  582
X36     EQU  584
X37     EQU  586
X38     EQU  588
X39     EQU  590
X40     EQU  592
X41     EQU  594
X42     EQU  596
X43     EQU  598
X44     EQU  600
X45     EQU  602
X46     EQU  604
X47     EQU  606
X48     EQU  608
X49     EQU  610
X50     EQU  612
X51     EQU  614
X52     EQU  616
X53     EQU  618
X54     EQU  620
X55     EQU  622
X56     EQU  624
X57     EQU  626
X58     EQU  628
X59     EQU  630
X60     EQU  632
X61     EQU  634
X62     EQU  636
X63     EQU  638
X64     EQU  640
X65     EQU  642
X66     EQU  644
X67     EQU  646
X68     EQU  648
X69     EQU  650
X70     EQU  652
X71     EQU  654
X72     EQU  656
X73     EQU  658
X74     EQU  660
X75     EQU  662
X76     EQU  664
X77     EQU  666
X78     EQU  668
X79     EQU  670
X80     EQU  672
X81     EQU  674
X82     EQU  676
X83     EQU  678
X84     EQU  680
X85     EQU  682
X86     EQU  684
X87     EQU  686
X88     EQU  688
X89     EQU  690
X90     EQU  692
X91     EQU  694
X92     EQU  696
X93     EQU  698
X94     EQU  700
X95     EQU  702
X96     EQU  704
X97     EQU  706
X98     EQU  708
X99     EQU  710
X100    EQU  712
X101    EQU  714
X102    EQU  716
X103    EQU  718
X104    EQU  720
X105    EQU  722
X106    EQU  724
X107    EQU  726
X108    EQU  728
X109    EQU  730
X110    EQU  732
X111    EQU  734
X112    EQU  736
X113    EQU  738
X114    EQU  740
X115    EQU  742
X116    EQU  744
X117    EQU  746
X118    EQU  748
X119    EQU  750
X120    EQU  752
X121    EQU  754
X122    EQU  756
X123    EQU  758
X124    EQU  760
X125    EQU  762
X126    EQU  764
X127    EQU  766
X128    EQU  768
X129    EQU  770
X130    EQU  772
X131    EQU  774
X132    EQU  776
X133    EQU  778
X134    EQU  780
X135    EQU  782
X136    EQU  784
X137    EQU  786
X138    EQU  788
X139    EQU  790
X140    EQU  792
X141    EQU  794
X142    EQU  796
X143    EQU  798
X144    EQU  800
X145    EQU  802
X146    EQU  804
X147    EQU  806
X148    EQU  808
X149    EQU  810
```

```
X150    EQU  812
X151    EQU  814
X152    EQU  816
X153    EQU  818
X154    EQU  820
X155    EQU  822
X156    EQU  824
X157    EQU  826
X158    EQU  828
X159    EQU  830
X160    EQU  832
X161    EQU  834
X162    EQU  836
X163    EQU  838
X164    EQU  840
X165    EQU  842
X166    EQU  844
X167    EQU  846
X168    EQU  848
X169    EQU  850
X170    EQU  852
X171    EQU  854
X172    EQU  856
X173    EQU  858
X174    EQU  860
X175    EQU  862
X176    EQU  864
X177    EQU  866
X178    EQU  868
X179    EQU  870
X180    EQU  872
X181    EQU  874
X182    EQU  876
X183    EQU  878
X184    EQU  880
X185    EQU  882
X186    EQU  884
X187    EQU  886
X188    EQU  888
X189    EQU  890
X190    EQU  892
X191    EQU  894
X192    EQU  896
X193    EQU  898
X194    EQU  900
X195    EQU  902
X196    EQU  904
X197    EQU  906
X198    EQU  908
X199    EQU  910
X200    EQU  912
X201    EQU  914
X202    EQU  916
X203    EQU  918
X204    EQU  920
X205    EQU  922
X206    EQU  924
X207    EQU  926
X208    EQU  928
X209    EQU  930
X210    EQU  932
X211    EQU  934
X212    EQU  936
X213    EQU  938
X214    EQU  940
X215    EQU  942
X216    EQU  944
X217    EQU  946
X218    EQU  948
X219    EQU  950
X220    EQU  952
X221    EQU  954
X222    EQU  956
X223    EQU  958
X224    EQU  960
X225    EQU  962
X226    EQU  964
X227    EQU  966
X228    EQU  968
X229    EQU  970
X230    EQU  972
X231    EQU  974
X232    EQU  976
X233    EQU  978
X234    EQU  980
X235    EQU  982
X236    EQU  984
X237    EQU  986
X238    EQU  988
X239    EQU  990
X240    EQU  992
X241    EQU  994
X242    EQU  996
X243    EQU  998
X244    EQU  1000
X245    EQU  1002
X246    EQU  1004
X247    EQU  1006
X248    EQU  1008
X249    EQU  1010
X250    EQU  1012
X251    EQU  1014
X252    EQU  1016
X253    EQU  1018
X254    EQU  1020
X255    EQU  1022
*
*  TABLE WITH COSINES
*
C0     EQU  4095
C1     EQU  4094
C2     EQU  4090
C3     EQU  4084
C4     EQU  4075
C5     EQU  4064
C6     EQU  4051
C7     EQU  4035
C8     EQU  4016
C9     EQU  3996
C10    EQU  3972
```

```
C11   EQU  3947
C12   EQU  3919
C13   EQU  3888
C14   EQU  3856
C15   EQU  3821
C16   EQU  3783
C17   EQU  3744
C18   EQU  3702
C19   EQU  3658
C20   EQU  3611
C21   EQU  3563
C22   EQU  3512
C23   EQU  3460
C24   EQU  3405
C25   EQU  3348
C26   EQU  3289
C27   EQU  3228
C28   EQU  3165
C29   EQU  3101
C30   EQU  3034
C31   EQU  2966
C32   EQU  2896
C33   EQU  2824
C34   EQU  2750
C35   EQU  2675
C36   EQU  2598
C37   EQU  2519
C38   EQU  2439
C39   EQU  2358
C40   EQU  2275
C41   EQU  2191
C42   EQU  2105
C43   EQU  2018
C44   EQU  1930
C45   EQU  1841
C46   EQU  1751
C47   EQU  1659
C48   EQU  1567
C49   EQU  1474
C50   EQU  1380
C51   EQU  1285
C52   EQU  1189
C53   EQU  1092
C54   EQU  995
C55   EQU  897
C56   EQU  799
C57   EQU  700
C58   EQU  601
C59   EQU  501
C60   EQU  401
C61   EQU  301
C62   EQU  201
C63   EQU  100
C64   EQU  0
C65   EQU  -99
C66   EQU  -200
C67   EQU  -300
C68   EQU  -400
C69   EQU  -500
C70   EQU  -600
C71   EQU  -699
C72   EQU  -798
C73   EQU  -896
C74   EQU  -994
C75   EQU  -1091
C76   EQU  -1188
C77   EQU  -1284
C78   EQU  -1379
C79   EQU  -1473
C80   EQU  -1566
C81   EQU  -1658
C82   EQU  -1750
C83   EQU  -1840
C84   EQU  -1929
C85   EQU  -2017
C86   EQU  -2104
C87   EQU  -2190
C88   EQU  -2274
C89   EQU  -2357
C90   EQU  -2438
C91   EQU  -2518
C92   EQU  -2597
C93   EQU  -2674
C94   EQU  -2749
C95   EQU  -2823
C96   EQU  -2895
C97   EQU  -2965
C98   EQU  -3033
C99   EQU  -3100
C100  EQU  -3164
C101  EQU  -3227
C102  EQU  -3288
C103  EQU  -3347
C104  EQU  -3404
C105  EQU  -3459
C106  EQU  -3511
C107  EQU  -3562
C108  EQU  -3610
C109  EQU  -3657
C110  EQU  -3701
C111  EQU  -3743
C112  EQU  -3782
C113  EQU  -3820
C114  EQU  -3855
C115  EQU  -3887
C116  EQU  -3918
C117  EQU  -3946
C118  EQU  -3971
C119  EQU  -3995
C120  EQU  -4015
C121  EQU  -4034
C122  EQU  -4050
C123  EQU  -4063
C124  EQU  -4074
C125  EQU  -4083
C126  EQU  -4089
C127  EQU  -4093
C128  EQU  -4094
C129  EQU  -4093
C130  EQU  -4089
```

```
C131  EQU  -4083
C132  EQU  -4074
C133  EQU  -4063
C134  EQU  -4050
C135  EQU  -4034
C136  EQU  -4015
C137  EQU  -3995
C138  EQU  -3971
C139  EQU  -3946
C140  EQU  -3918
C141  EQU  -3887
C142  EQU  -3855
C143  EQU  -3820
C144  EQU  -3782
C145  EQU  -3743
C146  EQU  -3701
C147  EQU  -3657
C148  EQU  -3610
C149  EQU  -3562
C150  EQU  -3511
C151  EQU  -3459
C152  EQU  -3404
C153  EQU  -3347
C154  EQU  -3288
C155  EQU  -3227
C156  EQU  -3164
C157  EQU  -3100
C158  EQU  -3033
C159  EQU  -2965
C160  EQU  -2895
C161  EQU  -2823
C162  EQU  -2749
C163  EQU  -2674
C164  EQU  -2597
C165  EQU  -2518
C166  EQU  -2438
C167  EQU  -2357
C168  EQU  -2274
C169  EQU  -2190
C170  EQU  -2104
C171  EQU  -2017
C172  EQU  -1929
C173  EQU  -1840
C174  EQU  -1750
C175  EQU  -1658
C176  EQU  -1566
C177  EQU  -1473
C178  EQU  -1379
C179  EQU  -1284
C180  EQU  -1188
C181  EQU  -1091
C182  EQU  -994
C183  EQU  -896
C184  EQU  -798
C185  EQU  -699
C186  EQU  -600
C187  EQU  -500
C188  EQU  -400
C189  EQU  -300
C190  EQU  -200
```

```
C191  EQU  -99
C192  EQU  0
C193  EQU  100
C194  EQU  201
C195  EQU  301
C196  EQU  401
C197  EQU  501
C198  EQU  601
C199  EQU  700
C200  EQU  799
C201  EQU  897
C202  EQU  995
C203  EQU  1092
C204  EQU  1189
C205  EQU  1285
C206  EQU  1380
C207  EQU  1474
C208  EQU  1567
C209  EQU  1659
C210  EQU  1751
C211  EQU  1841
C212  EQU  1930
C213  EQU  2018
C214  EQU  2105
C215  EQU  2191
C216  EQU  2275
C217  EQU  2358
C218  EQU  2439
C219  EQU  2519
C220  EQU  2598
C221  EQU  2675
C222  EQU  2750
C223  EQU  2824
C224  EQU  2896
C225  EQU  2966
C226  EQU  3034
C227  EQU  3101
C228  EQU  3165
C229  EQU  3228
C230  EQU  3289
C231  EQU  3348
C232  EQU  3405
C233  EQU  3460
C234  EQU  3512
C235  EQU  3563
C236  EQU  3611
C237  EQU  3658
C238  EQU  3702
C239  EQU  3744
C240  EQU  3783
C241  EQU  3821
C242  EQU  3856
C243  EQU  3888
C244  EQU  3919
C245  EQU  3947
C246  EQU  3972
C247  EQU  3996
C248  EQU  4016
C249  EQU  4035
C250  EQU  4051
```

```
C251  EQU  4064
C252  EQU  4075
C253  EQU  4084
C254  EQU  4090
C255  EQU  4094
*
*  TABLE WITH SINES
*
S0    EQU  0
S1    EQU  100
S2    EQU  201
S3    EQU  301
S4    EQU  401
S5    EQU  501
S6    EQU  601
S7    EQU  700
S8    EQU  799
S9    EQU  897
S10   EQU  995
S11   EQU  1092
S12   EQU  1189
S13   EQU  1285
S14   EQU  1380
S15   EQU  1474
S16   EQU  1567
S17   EQU  1659
S18   EQU  1751
S19   EQU  1841
S20   EQU  1930
S21   EQU  2018
S22   EQU  2105
S23   EQU  2191
S24   EQU  2275
S25   EQU  2358
S26   EQU  2439
S27   EQU  2519
S28   EQU  2598
S29   EQU  2675
S30   EQU  2750
S31   EQU  2824
S32   EQU  2896
S33   EQU  2966
S34   EQU  3034
S35   EQU  3101
S36   EQU  3165
S37   EQU  3228
S38   EQU  3289
S39   EQU  3348
S40   EQU  3405
S41   EQU  3460
S42   EQU  3512
S43   EQU  3563
S44   EQU  3611
S45   EQU  3658
S46   EQU  3702
S47   EQU  3744
S48   EQU  3783
S49   EQU  3821
S50   EQU  3856
S51   EQU  3888
S52   EQU  3919
S53   EQU  3947
S54   EQU  3972
S55   EQU  3996
S56   EQU  4016
S57   EQU  4035
S58   EQU  4051
S59   EQU  4064
S60   EQU  4075
S61   EQU  4084
S62   EQU  4090
S63   EQU  4094
S64   EQU  4095
S65   EQU  4094
S66   EQU  4090
S67   EQU  4084
S68   EQU  4075
S69   EQU  4064
S70   EQU  4051
S71   EQU  4035
S72   EQU  4016
S73   EQU  3996
S74   EQU  3972
S75   EQU  3947
S76   EQU  3919
S77   EQU  3888
S78   EQU  3856
S79   EQU  3821
S80   EQU  3783
S81   EQU  3744
S82   EQU  3702
S83   EQU  3658
S84   EQU  3611
S85   EQU  3563
S86   EQU  3512
S87   EQU  3460
S88   EQU  3405
S89   EQU  3348
S90   EQU  3289
S91   EQU  3228
S92   EQU  3165
S93   EQU  3101
S94   EQU  3034
S95   EQU  2966
S96   EQU  2896
S97   EQU  2824
S98   EQU  2750
S99   EQU  2675
S100  EQU  2598
S101  EQU  2519
S102  EQU  2439
S103  EQU  2358
S104  EQU  2275
S105  EQU  2191
S106  EQU  2105
S107  EQU  2018
S108  EQU  1930
S109  EQU  1841
S110  EQU  1751
S111  EQU  1659
```

```
S112  EQU  1567
S113  EQU  1474
S114  EQU  1380
S115  EQU  1285
S116  EQU  1189
S117  EQU  1092
S118  EQU  995
S119  EQU  897
S120  EQU  799
S121  EQU  700
S122  EQU  601
S123  EQU  501
S124  EQU  401
S125  EQU  301
S126  EQU  201
S127  EQU  100
S128  EQU  0
S129  EQU  -99
S130  EQU  -200
S131  EQU  -300
S132  EQU  -400
S133  EQU  -500
S134  EQU  -600
S135  EQU  -699
S136  EQU  -798
S137  EQU  -896
S138  EQU  -994
S139  EQU  -1091
S140  EQU  -1188
S141  EQU  -1284
S142  EQU  -1379
S143  EQU  -1473
S144  EQU  -1566
S145  EQU  -1658
S146  EQU  -1750
S147  EQU  -1840
S148  EQU  -1929
S149  EQU  -2017
S150  EQU  -2104
S151  EQU  -2190
S152  EQU  -2274
S153  EQU  -2357
S154  EQU  -2438
S155  EQU  -2518
S156  EQU  -2597
S157  EQU  -2674
S158  EQU  -2749
S159  EQU  -2823
S160  EQU  -2895
S161  EQU  -2965
S162  EQU  -3033
S163  EQU  -3100
S164  EQU  -3164
S165  EQU  -3227
S166  EQU  -3288
S167  EQU  -3347
S168  EQU  -3404
S169  EQU  -3459
S170  EQU  -3511
S171  EQU  -3562
S172  EQU  -3610
S173  EQU  -3657
S174  EQU  -3701
S175  EQU  -3743
S176  EQU  -3782
S177  EQU  -3820
S178  EQU  -3855
S179  EQU  -3887
S180  EQU  -3918
S181  EQU  -3946
S182  EQU  -3971
S183  EQU  -3995
S184  EQU  -4015
S185  EQU  -4034
S186  EQU  -4050
S187  EQU  -4063
S188  EQU  -4074
S189  EQU  -4083
S190  EQU  -4089
S191  EQU  -4093
S192  EQU  -4094
S193  EQU  -4093
S194  EQU  -4089
S195  EQU  -4083
S196  EQU  -4074
S197  EQU  -4063
S198  EQU  -4050
S199  EQU  -4034
S200  EQU  -4015
S201  EQU  -3995
S202  EQU  -3971
S203  EQU  -3946
S204  EQU  -3918
S205  EQU  -3887
S206  EQU  -3855
S207  EQU  -3820
S208  EQU  -3782
S209  EQU  -3743
S210  EQU  -3701
S211  EQU  -3657
S212  EQU  -3610
S213  EQU  -3562
S214  EQU  -3511
S215  EQU  -3459
S216  EQU  -3404
S217  EQU  -3347
S218  EQU  -3288
S219  EQU  -3227
S220  EQU  -3164
S221  EQU  -3100
S222  EQU  -3033
S223  EQU  -2965
S224  EQU  -2895
S225  EQU  -2823
S226  EQU  -2749
S227  EQU  -2674
S228  EQU  -2597
S229  EQU  -2518
S230  EQU  -2438
S231  EQU  -2357
```

```
S232  EQU  -2274
S233  EQU  -2190
S234  EQU  -2104
S235  EQU  -2017
S236  EQU  -1929
S237  EQU  -1840
S238  EQU  -1750
S239  EQU  -1658
S240  EQU  -1566
S241  EQU  -1473
S242  EQU  -1379
S243  EQU  -1284
S244  EQU  -1188
S245  EQU  -1091
S246  EQU  -994
S247  EQU  -896
S248  EQU  -798
S249  EQU  -699
S250  EQU  -600
S251  EQU  -500
S252  EQU  -400
S253  EQU  -300
S254  EQU  -200
S255  EQU  -99


      PAGE
*
*
*   COOLEY-TUKEY RADIX-4, DIF FFT PROGRAM - 256-POINT STRAIGHT-LINE.
*
*      FOUR STAGES OF THREE TYPES RADIX-4 BUTTERFLIES -
*      ZERO, SPCIAL, AND NORMAL - IMPLEMENTED WITH MACROS.
*      COMPLEX INPUT DATA LOCATED ON PAGES 4-7  OF DATA MEMORY.
*      RESULTS ARE LEFT IN DATA RAM.
*      USES 13-BIT COEFFICIENTS FROM MPYK INSTRUCTIONS.
*      INTERMEDIATE VALUES ARE SCALED BY 1/4 AT EACH STAGE TO PREVENT
*       OVERFLOW.
*
************************************************************************
*
* MACROS TO PRODUCE STRAIGHT-LINE 256-POINT COMPLEX FFT.
*
************************************************************************
*
*  ZERO FOR CASE W = 1 (THETA = 0).
*
*  X'S AND Y'S ARE INPUT AND OUTPUT LOCATIONS FOR BUTTERFLY.
*  ENTER WITH ARP = 1, AR1 --> XI1, AR2 --> XI, AR0 = [XI3]-[XI1]
*  EXIT WITH ARP = 1, AR1 --> NEXT XI1, AR2 --> NEXT XI
*
ZERO   $MACRO
*
       LAC  *0+,15             * (1/2)XI1
       ADD  *0-,15             * (1/2)XI3
       SACH *0+                * XI1 = (1/2)(XI1 + XI3)
       SUBH *                  * XI3
       SACH *0-,0,AR2          * XI3 = (1/2)(XI1 - XI3)
       LAC  *0+,14             * (1/4)XI
       SUB  *,14               * (1/4)XI2
       SACH T1                 * T1 = (1/4)(XI - XI2)
       ADD  *,15,AR1           * R1 (ACC) = (1/4)(XI + XI2)
       SUB  *,15,AR2           * (1/2)XI1
       SACH *0-,0,AR1          * XI2 = R1 - (1/2)XI1
       ADDH *+,AR2             * XI1
       SACH *+,0,AR1           * XI  = R1 + (1/2)XI1
*
       LAC  *0+,15             * (1/2)YI1
       ADD  *0-,15             * (1/2)YI3
       SACH *0+                * YI1 = (1/2)(YI1 + YI3)
       SUBH *                  * YI3
       SACH *0-,0,AR2          * YI3 = (1/2)(YI1 - YI3)
       LAC  *0+,14             * (1/4)YI
       SUB  *,14               * (1/4)YI2
       SACH T2                 * T2 = (1/4)(YI - YI2)
       ADD  *,15,AR1           * S1 (ACC) = (1/4)(YI + YI2)
       SUB  *,15,AR2           * (1/2)YI1
       SACH *0-,0,AR1          * YI2 = S1 - (1/2)YI1
       ADDH *0+,AR2            * YI1
       SACH *+,0,AR1           * YI  = S1 + (1/2)YI1, POINT TO NEXT XI
*
       ZALH T1
       ADD  *0-,15             * (1/2)YI3
       MAR  *-                 * POINT TO XI1
       SACH *+                 * XI1 = T1 + (1/2)YI3
       MAR  *0+                * POINT TO YI3
       SUBH *-                 * YI3
       SACH T1                 * T1 = T1 - (1/2)YI3
       ZALH T2
       ADD  *+,15              * (1/2)XI3
       SACH *-                 * YI3 = T2 + (1/2)XI3
       SUBH *                  * XI3
       LAR  AR3,T1             * AR3 = T1
       SAR  AR3,*+             * XI3 = T1
   .   MAR  *0-                * POINT TO YI1
       SACH *+                 * YI1 = T2 - (1/2)XI3 , POINT TO NEXT XI1
*
       $END
************************************************************************
*
*  NORMAL - STANDARD RADIX-4 BUTTERFLY.
*
*  X'S AND Y'S ARE INPUT AND OUTPUT LOCATIONS FOR BUTTERFLY.
*  IA'S SPECIFY TWIDDLE FACTOR LOCATIONS.
*  ENTER WITH ARP = 1, AR1 --> XI1, AR2 --> XI, AR0 = [XI3] - [XI1]
*  EXIT WITH ARP = 1, AR1 --> NEXT XI1, AR2 --> NEXT XI
*
NORMAL  $MACRO  IA1,IA2,IA3
*
       LAC  *0+,15             * (1/2)XI1
       ADD  *0-,15             * (1/2)XI3
       SACH *0+                * XI1 = (1/2)(XI1 + XI3)
       SUBH *                  * XI3
       SACH *0-,0,AR2          * XI3 = (1/2)(XI1 - XI3)
       LAC  *0+,14             * (1/4)XI
       SUB  *,14               * (1/4)XI2
       SACH T1                 * T1 = (1/4)(XI - XI2)
       ADD  *,15,AR1           * R1 (ACC) = (1/4)(XI + XI2)
       SUB  *,15,AR2           * (1/2)XI1
       SACH *0-,0,AR1          * XI2 = R1 - (1/2)XI1
```

```
ADDH *+,AR2          * XI1
SACH *+,0,AR1        * XI  = R1 + (1/2)XI1

LAC  *0+,15          * (1/2)YI1
ADD  *0-,15          * (1/2)YI3
SACH *0+             * YI1 = (1/2)(YI1 + YI3)
SUBH *               * YI3
SACH *0-,0,AR2       * YI3 = (1/2)(YI1 - YI3)
LAC  *0+,14          * (1/4)YI
SUB  *,14            * (1/4)YI2
SACH T2              * T2 = (1/4)(YI - YI2)
ADD  *,15,AR1        * S1 (ACC) = (1/4)(YI + YI2)
SUB  *,15,AR2        * (1/2)YI1
SACH *0-,0,AR1       * YI2 = S1 - (1/2)YI1
ADDH *0+,AR2         * YI1
SACH *0+,0,AR1       * YI  = S1 + (1/2)YI1, POINT TO YI2

ZALH T1
ADD  *0-,15          * (1/2)YI3
MAR  *-              * POINT TO XI1
SACH *+              * XI1 = T1 + (1/2)YI3
MAR  *0+             * POINT TO YI3
SUBH *-              * YI3
SACH T1              * T1 = T1 - (1/2)YI3
ZALH T2
ADD  *,15            * (1/2)XI3
SACH T2              * T2 = T2 + (1/2)XI3
SUBH *+              * XI3, POINT TO YI3
MAR  *0-             * POINT TO YI1
SACH *               * YI1 = T2 - (1/2)XI3

LT   *-              * YI1
MPYK C:IA1:          * CO1 * YI1
LTP  *+              * XI1
MPYK S:IA1:          * SI1 * XI1
LTS  *               * YI1
SACH *-,4            * YI1 = YI1*CO1 - XI1*SI1
MPYK S:IA1:          * SI1 * YI1
LTP  *,AR2           * XI1
MPYK C:IA1:          * CO1 * XI1
LTA  *-,AR1          * YI2
SACH *0+,4,AR2       * XI1 = YI1*SI1 + XI1*CO1, POINT TO XI3

MPYK C:IA2:          * CO2 * YI2
LTP  *+              * XI2
MPYK S:IA2:          * SI2 * XI2
LTS  *               * YI2
SACH *-,4            * YI2 = YI2*CO2 - XI2*SI2
MPYK S:IA2:          * SI2 * YI2
LTP  *               * XI2
MPYK C:IA2:          * CO2 * XI2
LTA  T1              * T1
SACH *0-,4           * XI2 = YI2*SI2 + XI2*CO2
MAR  *+,AR1

MPYK C:IA3:          * CO3 * T1
LTP  T2              * T2
MPYK S:IA3:          * SI3 * T2
APAC
SACH *+,4,AR2        * XI3 = T1*CO3 + T2*SI3

       MPYK C:IA3:          * CO3 * T2
       LTP  T1              * T1
       MPYK S:IA3:          * SI3 * T1
       LTS  *+,AR1          * SPAC, POINT TO NEXT XI, (LT = DUMMY OP)
       SACH *+,4            * YI3 = T2*CO3 - T1*SI3
       MAR  *0-             * POINT TO NEXT XI1
*
       $END
******************************************************************************
*
* SPCIAL FOR CASE THETA = pi/4
*
*  X'S AND Y'S ARE INPUT AND OUTPUT LOCATIONS FOR BUTTERFLY.
*  ENTER WITH ARP = 1, AR1 --> XI1, AR2 --> XI, AR0 = [XI3] - [XI1]
*  EXIT WITH ARP = 1, AR1 --> NEXT XI1, AR2 --> NEXT XI
*
SPCIAL $MACRO
*
       LAC  *0+,14          * (1/4)XI1
       ADD  *0-,14          * (1/4)XI3
       SACH *0+             * XI1 = (1/4)(XI1 + XI3)
       SUB  *,15            * (1/2)XI3
       SACH *+              * XI3 = (1/4)(XI1 - XI3)
*
       LAC  *0-,14          * (1/4)YI3
       ADD  *,14            * (1/4)YI1
       SACH *0+             * YI1 = (1/4)(YI1 + YI3)
       SUB  *,15            * (1/2)YI3
       SACH *0-,0,AR2       * YI3 = (1/4)(YI1 - YI3), POINT TO YI1
*
       LAC  *0+,14          * (1/4)XI
       ADD  *0-,14          * (1/4)XI2
       SACH *0+             * XI = (1/4)(XI + XI2)
       SUB  *,15            * (1/2)XI2
       SACH *+              * XI2 = (1/4)(XI - XI2)
*
       LAC  *0-,14          * (1/4)YI2
       ADD  *,14            * (1/4)YI
       SACH *0+             * YI = (1/4)(YI + YI2)
       SUB  *,15            * (1/2)YI2
       SACH *0-             * YI2 = (1/4)(YI - YI2), POINT TO YI
*
       LAC  *,0,AR1         * YI
       ADD  *,0,AR2         * YI1
       SACL *-,0,AR1        * YI = YI + YI1
       SUB  *-,1            * 2 YI1
       SACL T2              * T2 = YI - YI1
       LAC  *0+,0,AR2       * XI1, POINT TO XI3
       SUB  *               * XI
       SACL T1              * T1 = XI1 - XI
       ADD  *,1             * 2 XI
       SACL *0+             * XI = XI1 + XI, POINT TO XI2
*
       LT   *+,AR1          * XI2
       MPYK C32
       MAR  *+
       LTP  *               * YI3
       MPYK C32
       SPAC
       SACH *-,4            * YI3 = (XI2-YI3)*C32
```

```
        APAC
        LTA  *0-                * XI3
        SACH *0+,4,AR2          * XI1 = (XI2+YI3)*C32
*
        MPYK C32
        LTP  *,AR1              * YI2
        MPYK C32
        APAC
        SACH *0-,4              * XI3 = (YI2+XI3)*C32
        SPAC
        SPAC
        NEG
        MAR  *+
        SACH *,4,AR2            * YI1 = (YI2-XI3)*C32
*
        LAC  T1
        SACL *-                 * YI2 = T1
        LAC  T2
        SACL *0-                * XI2 = T2
        MAR  *+
        MAR  *+,AR1             * POINT TO NEXT XI
*
        LAC  *-                 * YI1
        SUB  *+                 * XI1
        SACL *-                 * YI1 = YI1 - XI1
        ADD  *,1                * 2 XI1
        SACL *0+                * XI1 = YI1 + XI1
        LAC  *+                 * XI3
        SUB  *-                 * YI3
        SACL *+                 * XI3 = XI3 - YI3
        ADD  *,1                * 2 YI3
        NEG
        SACL *0-                * YI3 = -(YI3 + XI3)
        MAR  *+                 * POINT TO NEXT XI1
*
        $END
*****************************************************************************
*
* DIGREV MACRO TO DO EXCHANGE OF LOCATIONS FOR DIGIT REVERSAL.
*
DIGREV $MACRO I,J
*
        LRLK AR1,X:I:           * AR1 POINTS TO XI
        LRLK AR2,X:J:           * AR1 POINTS TO XJ
*
        ZALH *,AR2
        ADDS *,AR1
        SACL *+,0,AR2
        SACH *+,0,AR1
        ZALH *,AR2
        ADDS *,AR1
        SACL *,0,AR2
        SACH *,0,AR1
*
        $END
*
*****************************************************************************
*
*       MAIN ROUTINE TO CALL ABOVE MACROS WITH APPROPRIATE PARAMETERS.
*
*****************************************************************************
      AORG 0
      B    32
*
      AORG 32
FF256 LDPK 0                     * STORE ALL TEMPORARY VARIABLES IN B2.
      SOVM
      SSXM
      SPM  0
      LARP 1
*
*        INPUT DATA (256 REAL AND 256 IMAGINARY VALUES)
*
      LRLK AR1,X0
      RPTK 255
      IN   *+,PA0
      RPTK 255
      IN   *+,PA0
*****************************************************************************
*
* PASS 1
*
*****************************************************************************
      LRLK    AR0,256           * STEP VALUE BETWEEN BUTTERFLY "LEGS"
      LRLK    AR1,X64
      LRLK    AR2,X0
      ZERO                      * X0,X64,X128,X192,Y0,Y64,Y128,Y192
*
      NORMAL  1,2,3             * X1,X65,X129,X193,Y1,Y65,Y129,Y193
      NORMAL  2,4,6             * X2,X66,X130,X194,...
      NORMAL  3,6,9
      NORMAL  4,8,12
      NORMAL  5,10,15
      NORMAL  6,12,18
      NORMAL  7,14,21
      NORMAL  8,16,24
      NORMAL  9,18,27
      NORMAL  10,20,30
      NORMAL  11,22,33
      NORMAL  12,24,36
      NORMAL  13,26,39
      NORMAL  14,28,42
      NORMAL  15,30,45
      NORMAL  16,32,48
      NORMAL  17,34,51
      NORMAL  18,36,54
      NORMAL  19,38,57
      NORMAL  20,40,60
      NORMAL  21,42,63
      NORMAL  22,44,66
      NORMAL  23,46,69
      NORMAL  24,48,72
      NORMAL  25,50,75
      NORMAL  26,52,78
```

```
       NORMAL  27,54,81
       NORMAL  28,56,84
       NORMAL  29,58,87
       NORMAL  30,60,90
       NORMAL  31,62,93           * X31,X95,X159,X223,Y31,Y95,Y159,Y223
*
       SPCIAL                     * X32,X96,X160,X224,Y32,Y96,Y160,Y224
*
       NORMAL  33,66,99           * X33,X97,X161,X225,Y33,Y97,Y161,Y225
       NORMAL  34,68,102          * X34,X98,X162,X226,...
       NORMAL  35,70,105
       NORMAL  36,72,108
       NORMAL  37,74,111
       NORMAL  38,76,114
       NORMAL  39,78,117
       NORMAL  40,80,120
       NORMAL  41,82,123
       NORMAL  42,84,126
       NORMAL  43,86,129
       NORMAL  44,88,132
       NORMAL  45,90,135
       NORMAL  46,92,138
       NORMAL  47,94,141
       NORMAL  48,96,144
       NORMAL  49,98,147
       NORMAL  50,100,150
       NORMAL  51,102,153
       NORMAL  52,104,156
       NORMAL  53,106,159
       NORMAL  54,108,162
       NORMAL  55,110,165
       NORMAL  56,112,168
       NORMAL  57,114,171
       NORMAL  58,116,174
       NORMAL  59,118,177
       NORMAL  60,120,180
       NORMAL  61,122,183
       NORMAL  62,124,186
       NORMAL  63,126,189         * X63,X127,X191,X255,Y63,Y127,Y191,Y255
*
*******************************************************************************
*
* PASS 2
*
*******************************************************************************
*
*  FIRST SET OF BUTTERFLIES
*
       LARK    AR0,64             * STEP VALUE BETWEEN BUTTERFLY "LEGS"
       LRLK    AR1,X16
       LRLK    AR2,X0
       ZERO                       * X0,X16,X32,X48,Y0,Y16,Y32,Y48
*
       NORMAL  4,8,12             * X1,X17,X33,X49,Y1,Y17,Y33,Y49
       NORMAL  8,16,24            * X2,X18,X34,X50,...
       NORMAL  12,24,36
       NORMAL  16,32,48
       NORMAL  20,40,60
       NORMAL  24,48,72
       NORMAL  28,56,84           * X7,X23,X39,X55,Y7,Y23,Y39,Y55
*
       SPCIAL                     * X8,X24,X40,X56,Y8,Y24,Y40,Y56
*
       NORMAL  36,72,108          * X9,X25,X41,X57,Y9,Y25,Y41,Y57
       NORMAL  40,80,120          * X10,X26,X42,X58,...
       NORMAL  44,88,132
       NORMAL  48,96,144
       NORMAL  52,104,156
       NORMAL  56,112,168
       NORMAL  60,120,180         * X15,X31,X47,X63,Y15,Y31,Y47,Y63
*
*  SECOND SET OF BUTTERFLIES
*
       LRLK    AR1,X80
       LRLK    AR2,X64
       ZERO                       * X64,X80,X96,X112,Y64,Y80,Y96,Y112
*
       NORMAL  4,8,12             * X65,X81,X97,X113,Y65,Y81,Y97,Y113
       NORMAL  8,16,24            * X66,X82,X98,X114,...
       NORMAL  12,24,36
       NORMAL  16,32,48
       NORMAL  20,40,60
       NORMAL  24,48,72
       NORMAL  28,56,84           * X71,X87,X103,X119,Y71,Y87,Y103,Y119
*
       SPCIAL                     * X72,X88,X104,X120,Y72,Y88,Y104,Y120
*
       NORMAL  36,72,108          * X73,X89,X105,X121,Y73,Y89,Y105,Y121
       NORMAL  40,80,120          * X74,X90,X106,X122,...
       NORMAL  44,88,132
       NORMAL  48,96,144
       NORMAL  52,104,156
       NORMAL  56,112,168
       NORMAL  60,120,180         * X79,X95,X111,X127,Y79,Y95,Y111,Y127
*
*  THIRD SET OF BUTTERFLIES
*
       LRLK    AR1,X144
       LRLK    AR2,X128
       ZERO                       * X128,X144,X160,X176,Y128,Y144,Y160,Y176
*
       NORMAL  4,8,12             * X129,X145,X161,X177,Y129,Y145,Y161,Y177
       NORMAL  8,16,24            * X130,X146,X162,X178,...
       NORMAL  12,24,36
       NORMAL  16,32,48
       NORMAL  20,40,60
       NORMAL  24,48,72
       NORMAL  28,56,84           * X135,X151,X167,X183,Y135,Y151,Y167,Y183
*
       SPCIAL                     * X136,X152,X168,X184,Y136,Y152,Y168,Y184
*
       NORMAL  36,72,108          * X137,X153,X169,X185,Y137,Y153,Y169,Y185
       NORMAL  40,80,120          * X138,X154,X170,X186,...
       NORMAL  44,88,132
       NORMAL  48,96,144
       NORMAL  52,104,156
       NORMAL  56,112,168
       NORMAL  60,120,180         * X143,X159,X175,X191,Y143,Y159,Y175,Y191
*
*  FOURTH SET OF BUTTERFLIES
```

```
*
       LRLK    AR1,X208
       LRLK    AR2,X192
       ZERO                     * X192,X208,X224,X240,Y192,Y208,Y224,Y240
*
       NORMAL  4,8,12           * X193,X209,X225,X241,Y193,Y209,Y225,Y241
       NORMAL  8,16,24          * X194,X210,X226,X242,...
       NORMAL  12,24,36
       NORMAL  16,32,48
       NORMAL  20,40,60
       NORMAL  24,48,72
       NORMAL  28,56,84         * X199,X215,X231,X247,Y199,Y215,Y231,Y247
*
       SPCIAL                   * X200,X216,X232,X248,Y200,Y216,Y232,Y248
*
       NORMAL  36,72,108        * X201,X217,X233,X249,Y201,Y217,Y233,Y249
       NORMAL  40,80,120        * X202,X218,X234,X250,...
       NORMAL  44,88,132
       NORMAL  48,96,144
       NORMAL  52,104,156
       NORMAL  56,112,168
       NORMAL  60,120,180       * X207,X223,X239,X255,Y207,Y223,Y239,Y255
*
*
********************************************************************************
*
* PASS 3
*
********************************************************************************
*
       LARK    AR0,16           * STEP VALUE BETWEEN BUTTERFLY "LEGS"
*
       LRLK    AR1,X4
       LRLK    AR2,X0
*
       ZERO                     * X0,X4,X8,X12,Y0,Y4,Y8,Y12
       NORMAL  16,32,48         * X1,X5,X9,X13,Y1,Y5,Y9,Y13
       SPCIAL                   * X2,X6,X10,X14,Y2,Y6,Y10,Y14
       NORMAL  48,96,144        * X3,X7,X11,X15,Y3,Y7,Y11,Y15
*
       LRLK    AR1,X20
       LRLK    AR2,X16
*
       ZERO                     * X16,X20,X24,X28,Y16,Y20,Y24,Y28
       NORMAL  16,32,48         * X17,X21,X25,X29,Y17,Y21,Y25,Y29
       SPCIAL                   * X18,X22,X26,X30,Y18,Y22,Y26,Y30
       NORMAL  48,96,144        * X19,X23,X27,X31,Y19,Y23,Y27,Y31
*
       LRLK    AR1,X36
       LRLK    AR2,X32
*
       ZERO                     * X32,X36,X40,X44,Y32,Y36,Y40,Y44
       NORMAL  16,32,48         * X33,X37,X41,X45,Y33,Y37,Y41,Y45
       SPCIAL                   * X34,X38,X42,X46,Y34,Y38,Y42,Y46
       NORMAL  48,96,144        * X35,X39,X43,X47,Y35,Y39,Y43,Y47
*
       LRLK    AR1,X52
       LRLK    AR2,X48
*
       ZERO                     * X48,X52,X56,X60,Y48,Y52,Y56,Y60
       NORMAL  16,32,48         * X49,X53,X57,X61,Y49,Y53,Y57,Y61
       SPCIAL                   * X50,X54,X58,X62,Y50,Y54,Y58,Y62
       NORMAL  48,96,144        * X51,X55,X59,X63,Y51,Y55,Y59,Y63
*
       LRLK    AR1,X68
       LRLK    AR2,X64
*
       ZERO                     * X64,X68,X72,X76,Y64,Y68,Y72,Y76
       NORMAL  16,32,48         * X65,X69,X73,X77,Y65,Y69,Y73,Y77
       SPCIAL                   * X66,X70,X74,X78,Y66,Y70,Y74,Y78
       NORMAL  48,96,144        * X67,X71,X75,X79,Y67,Y71,Y75,Y79
*
       LRLK    AR1,X84
       LRLK    AR2,X80
*
       ZERO                     * X80,X84,X88,X92,Y80,Y84,Y88,Y92
       NORMAL  16,32,48         * X81,X85,X89,X93,Y81,Y85,Y89,Y93
       SPCIAL                   * X82,X86,X90,X94,Y82,Y86,Y90,Y94
       NORMAL  48,96,144        * X83,X87,X91,X95,Y83,Y87,Y91,Y95
*
       LRLK    AR1,X100
       LRLK    AR2,X96
*
       ZERO                     * X96,X100,X104,X108,Y96,Y100,Y104,Y108
       NORMAL  16,32,48         * X97,X101,X105,X109,Y97,Y101,Y105,Y109
       SPCIAL                   * X98,X102,X106,X110,Y98,Y102,Y106,Y110
       NORMAL  48,96,144        * X99,X103,X107,X111,Y99,Y103,Y107,Y111
*
       LRLK    AR1,X116
       LRLK    AR2,X112
*
       ZERO                     * X112,X116,X120,X124,Y112,Y116,Y120,Y124
       NORMAL  16,32,48         * X113,X117,X121,X125,Y113,Y117,Y121,Y125
       SPCIAL                   * X114,X118,X122,X126,Y114,Y118,Y122,Y126
       NORMAL  48,96,144        * X115,X119,X123,X127,Y115,Y119,Y123,Y127
*
       LRLK    AR1,X132
       LRLK    AR2,X128
*
       ZERO                     * X128,X132,X136,X140,Y128,Y132,Y136,Y140
       NORMAL  16,32,48         * X129,X133,X137,X141,Y129,Y133,Y137,Y141
       SPCIAL                   * X130,X134,X138,X142,Y130,Y134,Y138,Y142
       NORMAL  48,96,144        * X131,X135,X139,X143,Y131,Y135,Y139,Y143
*
       LRLK    AR1,X148
       LRLK    AR2,X144
*
       ZERO                     * X144,X148,X152,X156,Y144,Y148,Y152,Y156
       NORMAL  16,32,48         * X145,X149,X153,X157,Y145,Y149,Y153,Y157
       SPCIAL                   * X146,X150,X154,X158,Y146,Y150,Y154,Y158
       NORMAL  48,96,144        * X147,X151,X155,X159,Y147,Y151,Y155,Y159
*
       LRLK    AR1,X164
       LRLK    AR2,X160
*
       ZERO                     * X160,X164,X168,X172,Y160,Y164,Y168,Y172
       NORMAL  16,32,48         * X161,X165,X169,X173,Y161,Y165,Y169,Y173
       SPCIAL                   * X162,X166,X170,X174,Y162,Y166,Y170,Y174
       NORMAL  48,96,144        * X163,X167,X171,X175,Y163,Y167,Y171,Y175
*
```

```
        LRLK    AR1,X180
        LRLK    AR2,X176
*
        ZERO                    * X176,X180,X184,X188,Y176,Y180,Y184,Y188
        NORMAL  16,32,48        * X177,X181,X185,X189,Y177,Y181,Y185,Y189
        SPCIAL                  * X178,X182,X186,X190,Y178,Y182,Y186,Y190
        NORMAL  48,96,144       * X179,X183,X187,X191,Y179,Y183,Y187,Y191
*
        LRLK    AR1,X196
        LRLK    AR2,X192
*
        ZERO                    * X192,X196,X200,X204,Y192,Y196,Y200,Y204
        NORMAL  16,32,48        * X193,X197,X201,X205,Y193,Y197,Y201,Y205
        SPCIAL                  * X194,X198,X202,X206,Y194,Y198,Y202,Y206
        NORMAL  48,96,144       * X195,X199,X203,X207,Y195,Y199,Y203,Y207
*
        LRLK    AR1,X212
        LRLK    AR2,X208
*
        ZERO                    * X208,X212,X216,X220,Y208,Y212,Y216,Y220
        NORMAL  16,32,48        * X209,X213,X217,X221,Y209,Y213,Y217,Y221
        SPCIAL                  * X210,X214,X218,X222,Y210,Y214,Y218,Y222
        NORMAL  48,96,144       * X211,X215,X219,X223,Y211,Y215,Y219,Y223
*
        LRLK    AR1,X228
        LRLK    AR2,X224
*
        ZERO                    * X224,X228,X232,X236,Y224,Y228,Y232,Y236
        NORMAL  16,32,48        * X225,X229,X233,X237,Y225,Y229,Y233,Y237
        SPCIAL                  * X226,X230,X234,X238,Y226,Y230,Y234,Y238
        NORMAL  48,96,144       * X227,X231,X235,X239,Y227,Y231,Y235,Y239
*
        LRLK    AR1,X244
        LRLK    AR2,X240
*
        ZERO                    * X240,X244,X248,X252,Y240,Y244,Y248,Y252
        NORMAL  16,32,48        * X241,X245,X249,X253,Y241,Y245,Y249,Y253
        SPCIAL                  * X242,X246,X250,X254,Y242,Y246,Y250,Y254
        NORMAL  48,96,144       * X243,X247,X251,X255,Y243,Y247,Y251,Y255
*
*
************************************************************************************
*
* PASS 4
*
************************************************************************************
*
        LARK    AR0,4           * STEP VALUE BETWEEN BUTTERFLY "LEGS"
*
        LRLK    AR1,X1
        LRLK    AR2,X0
        ZERO                    * X0,X1,X2,X3,Y0,Y1,Y2,Y3
        LRLK    AR1,X5
        LRLK    AR2,X4
        ZERO                    * X4,X5,X6,X7,Y4,Y5,Y6,Y7
        LRLK    AR1,X9
        LRLK    AR2,X8
        ZERO                    * X8,X9,X10,X11,Y8,Y9,Y10,Y11
        LRLK    AR1,X13
        LRLK    AR2,X12
        ZERO                    * X12,X13,X14,X15,Y12,Y13,Y14,Y15
        LRLK    AR1,X17
        LRLK    AR2,X16
        ZERO                    * X16,X17,X18,X19,Y16,Y17,Y18,Y19
        LRLK    AR1,X21
        LRLK    AR2,X20
        ZERO                    * X20,X21,X22,X23,Y20,Y21,Y22,Y23
        LRLK    AR1,X25
        LRLK    AR2,X24
        ZERO                    * X24,X25,X26,X27,Y24,Y25,Y26,Y27
        LRLK    AR1,X29
        LRLK    AR2,X28
        ZERO                    * X28,X29,X30,X31,Y28,Y29,Y30,Y31
        LRLK    AR1,X33
        LRLK    AR2,X32
        ZERO                    * X32,X33,X34,X35,Y32,Y33,Y34,Y35
        LRLK    AR1,X37
        LRLK    AR2,X36
        ZERO                    * X36,X37,X38,X39,Y36,Y37,Y38,Y39
        LRLK    AR1,X41
        LRLK    AR2,X40
        ZERO                    * X40,X41,X42,X43,Y40,Y41,Y42,Y43
        LRLK    AR1,X45
        LRLK    AR2,X44
        ZERO                    * X44,X45,X46,X47,Y44,Y45,Y46,Y47
        LRLK    AR1,X49
        LRLK    AR2,X48
        ZERO                    * X48,X49,X50,X51,Y48,Y49,Y50,Y51
        LRLK    AR1,X53
        LRLK    AR2,X52
        ZERO                    * X52,X53,X54,X55,Y52,Y53,Y54,Y55
        LRLK    AR1,X57
        LRLK    AR2,X56
        ZERO                    * X56,X57,X58,X59,Y56,Y57,Y58,Y59
        LRLK    AR1,X61
        LRLK    AR2,X60
        ZERO                    * X60,X61,X62,X63,Y60,Y61,Y62,Y63
*
        LRLK    AR1,X65
        LRLK    AR2,X64
        ZERO                    * X64,X65,X66,X67,Y64,Y65,Y66,Y67
        LRLK    AR1,X69
        LRLK    AR2,X68
        ZERO                    * X68,X69,X70,X71,Y68,Y69,Y70,Y71
        LRLK    AR1,X73
        LRLK    AR2,X72
        ZERO                    * X72,X73,X74,X75,Y72,Y73,Y74,Y75
        LRLK    AR1,X77
        LRLK    AR2,X76
        ZERO                    * X76,X77,X78,X79,Y76,Y77,Y78,Y79
        LRLK    AR1,X81
        LRLK    AR2,X80
        ZERO                    * X80,X81,X82,X83,Y80,Y81,Y82,Y83
        LRLK    AR1,X85
        LRLK    AR2,X84
        ZERO                    * X84,X85,X86,X87,Y84,Y85,Y86,Y87
        LRLK    AR1,X89
        LRLK    AR2,X88
        ZERO                    * X88,X89,X90,X91,Y88,Y89,Y90,Y91
        LRLK    AR1,X93
```

```
LRLK    AR2,X92
ZERO                  * X92,X93,X94,X95,Y92,Y93,Y94,Y95
LRLK    AR1,X97
LRLK    AR2,X96
ZERO                  * X96,X97,X98,X99,Y96,Y97,Y98,Y99
LRLK    AR1,X101
LRLK    AR2,X100
ZERO                  * X100,X101,X102,X103,Y100,Y101,Y102,Y103
LRLK    AR1,X105
LRLK    AR2,X104
ZERO                  * X104,X105,X106,X107,Y104,Y105,Y106,Y107
LRLK    AR1,X109
LRLK    AR2,X108
ZERO                  * X108,X109,X110,X111,Y108,Y109,Y110,Y111
LRLK    AR1,X113
LRLK    AR2,X112
ZERO                  * X112,X113,X114,X115,Y112,Y113,Y114,Y115
LRLK    AR1,X117
LRLK    AR2,X116
ZERO                  * X116,X117,X118,X119,Y116,Y117,Y118,Y119
LRLK    AR1,X121
LRLK    AR2,X120
ZERO                  * X120,X121,X122,X123,Y120,Y121,Y122,Y123
LRLK    AR1,X125
LRLK    AR2,X124
ZERO                  * X124,X125,X126,X127,Y124,Y125,Y126,Y127

LRLK    AR1,X129
LRLK    AR2,X128
ZERO                  * X128,X129,X130,X131,Y128,Y129,Y130,Y131
LRLK    AR1,X133
LRLK    AR2,X132
ZERO                  * X132,X133,X134,X135,Y132,Y133,Y134,Y135
LRLK    AR1,X137
LRLK    AR2,X136
ZERO                  * X136,X137,X138,X139,Y136,Y137,Y138,Y139
LRLK    AR1,X141
LRLK    AR2,X140
ZERO                  * X140,X141,X142,X143,Y140,Y141,Y142,Y143
LRLK    AR1,X145
LRLK    AR2,X144
ZERO                  * X144,X145,X146,X147,Y144,Y145,Y146,Y147
LRLK    AR1,X149
LRLK    AR2,X148
ZERO                  * X148,X149,X150,X151,Y148,Y149,Y150,Y151
LRLK    AR1,X153
LRLK    AR2,X152
ZERO                  * X152,X153,X154,X155,Y152,Y153,Y154,Y155
LRLK    AR1,X157
LRLK    AR2,X156
ZERO                  * X156,X157,X158,X159,Y156,Y157,Y158,Y159
LRLK    AR1,X161
LRLK    AR2,X160
ZERO                  * X160,X161,X162,X163,Y160,Y161,Y162,Y163
LRLK    AR1,X165
LRLK    AR2,X164
ZERO                  * X164,X165,X166,X167,Y164,Y165,Y166,Y167
LRLK    AR1,X169
LRLK    AR2,X168
ZERO                  * X168,X169,X170,X171,Y168,Y169,Y170,Y171
LRLK    AR1,X173
LRLK    AR2,X172
ZERO                  * X172,X173,X174,X175,Y172,Y173,Y174,Y175
LRLK    AR1,X177
LRLK    AR2,X176
ZERO                  * X176,X177,X178,X179,Y176,Y177,Y178,Y179
LRLK    AR1,X181
LRLK    AR2,X180
ZERO                  * X180,X181,X182,X183,Y180,Y181,Y182,Y183
LRLK    AR1,X185
LRLK    AR2,X184
ZERO                  * X184,X185,X186,X187,Y184,Y185,Y186,Y187
LRLK    AR1,X189
LRLK    AR2,X188
ZERO                  * X188,X189,X190,X191,Y188,Y189,Y190,Y191
*
LRLK    AR1,X193
LRLK    AR2,X192
ZERO                  * X192,X193,X194,X195,Y192,Y193,Y194,Y195
LRLK    AR1,X197
LRLK    AR2,X196
ZERO                  * X196,X197,X198,X199,Y196,Y197,Y198,Y199
LRLK    AR1,X201
LRLK    AR2,X200
ZERO                  * X200,X201,X202,X203,Y200,Y201,Y202,Y203
LRLK    AR1,X205
LRLK    AR2,X204
ZERO                  * X204,X205,X206,X207,Y204,Y205,Y206,Y207
LRLK    AR1,X209
LRLK    AR2,X208
ZERO                  * X208,X209,X210,X211,Y208,Y209,Y210,Y211
LRLK    AR1,X213
LRLK    AR2,X212
ZERO                  * X212,X213,X214,X215,Y212,Y213,Y214,Y215
LRLK    AR1,X217
LRLK    AR2,X216
ZERO                  * X216,X217,X218,X219,Y216,Y217,Y218,Y219
LRLK    AR1,X221
LRLK    AR2,X220
ZERO                  * X220,X221,X222,X223,Y220,Y221,Y222,Y223
LRLK    AR1,X225
LRLK    AR2,X224
ZERO                  * X224,X225,X226,X227,Y224,Y225,Y226,Y227
LRLK    AR1,X229
LRLK    AR2,X228
ZERO                  * X228,X229,X230,X231,Y228,Y229,Y230,Y231
LRLK    AR1,X233
LRLK    AR2,X232
ZERO                  * X232,X233,X234,X235,Y232,Y233,Y234,Y235
LRLK    AR1,X237
LRLK    AR2,X236
ZERO                  * X236,X237,X238,X239,Y236,Y237,Y238,Y239
LRLK    AR1,X241
LRLK    AR2,X240
ZERO                  * X240,X241,X242,X243,Y240,Y241,Y242,Y243
LRLK    AR1,X245
LRLK    AR2,X244
ZERO                  * X244,X245,X246,X247,Y244,Y245,Y246,Y247
LRLK    AR1,X249
LRLK    AR2,X248
```

```
        ZERO                    * X248,X249,X250,X251,Y248,Y249,Y250,Y251
        LRLK    AR1,X253
        LRLK    AR2,X252
        ZERO                    * X252,X253,X254,X255,Y252,Y253,Y254,Y255
*
*
* DIGIT REVERSE COUNTER FOR RADIX-4 FFT COMPUTATION.
*
          LARP 1
        DIGREV  1,64
        DIGREV  2,128
        DIGREV  3,192
        DIGREV  4,16
        DIGREV  5,80
        DIGREV  6,144
        DIGREV  7,208
        DIGREV  8,32
        DIGREV  9,96
        DIGREV  10,160
        DIGREV  11,224
        DIGREV  12,48
        DIGREV  13,112
        DIGREV  14,176
        DIGREV  15,240
        DIGREV  17,68
        DIGREV  18,132
        DIGREV  19,196
        DIGREV  21,84
        DIGREV  22,148
        DIGREV  23,212
        DIGREV  24,36
        DIGREV  25,100
        DIGREV  26,164
        DIGREV  27,228
        DIGREV  28,52
        DIGREV  29,116
        DIGREV  30,180
        DIGREV  31,244
        DIGREV  33,72
        DIGREV  34,136
        DIGREV  35,200
        DIGREV  37,88
        DIGREV  38,152
        DIGREV  39,216
        DIGREV  41,104
        DIGREV  42,168
        DIGREV  43,232
        DIGREV  44,56
        DIGREV  45,120
        DIGREV  46,184
        DIGREV  47,248
        DIGREV  49,76
        DIGREV  50,140
        DIGREV  51,204
        DIGREV  53,92
        DIGREV  54,156
        DIGREV  55,220
        DIGREV  57,108
        DIGREV  58,172
        DIGREV  59,236
        DIGREV  61,124
        DIGREV  62,188
        DIGREV  63,252
        DIGREV  66,129
        DIGREV  67,193
        DIGREV  69,81
        DIGREV  70,145
        DIGREV  71,209
        DIGREV  73,97
        DIGREV  74,161
        DIGREV  75,225
        DIGREV  77,113
        DIGREV  78,177
        DIGREV  79,241
        DIGREV  82,133
        DIGREV  83,197
        DIGREV  86,149
        DIGREV  87,213
        DIGREV  89,101
        DIGREV  90,165
        DIGREV  91,229
        DIGREV  93,117
        DIGREV  94,181
        DIGREV  95,245
        DIGREV  98,137
        DIGREV  99,201
        DIGREV  102,153
        DIGREV  103,217
        DIGREV  106,169
        DIGREV  107,233
        DIGREV  109,121
        DIGREV  110,185
        DIGREV  111,249
        DIGREV  114,141
        DIGREV  115,205
        DIGREV  118,157
        DIGREV  119,221
        DIGREV  122,173
        DIGREV  123,237
        DIGREV  126,189
        DIGREV  127,253
        DIGREV  131,194
        DIGREV  134,146
        DIGREV  135,210
        DIGREV  138,162
        DIGREV  139,226
        DIGREV  142,178
        DIGREV  143,242
        DIGREV  147,198
        DIGREV  151,214
        DIGREV  154,166
        DIGREV  155,230
        DIGREV  158,182
        DIGREV  159,246
        DIGREV  163,202
        DIGREV  167,218
        DIGREV  171,234
        DIGREV  174,186
        DIGREV  175,250
        DIGREV  179,206
        DIGREV  183,222
        DIGREV  187,238
        DIGREV  191,254
        DIGREV  199,211
        DIGREV  203,227
        DIGREV  207,243
        DIGREV  219,231
        DIGREV  223,247
        DIGREV  239,251
*
*       OUTPUT FFT DATA
*
        LARP 1
        LRLK AR1,X0
        RPTK 255
        OUT  *+,PA1
        RPTK 255
        OUT  *+,PA1
*
*  FFT COMPLETE.
*
WHOA    B    WHOA
        END
```

APPENDIX F
A 128-POINT, RADIX-2 DIF FFT IMPLEMENTATION (LOOPED CODE)

```
        IDT 'FFT2'
*
*    COOLEY-TUKEY 128-POINT, RADIX-2, DIF FFT PROGRAM FOR THE TMS32020.
*
*       SINGLE FFT BUTTERFLY.
*       REAL INPUT DATA, STORED IN PAGE 4 OF BLOCK B0.
*       FFT COMPUTATION IS DONE IN PAGES 6, 7 OF BLOCK B1 (COMPLEX NUMBERS).
*       USES TABLE LOOKUP (FROM PROGRAM MEMORY) OF THE TWIDDLE FACTORS.
*       INTERMEDIATE VALUES ARE SCALED BY .5 AT EACH STAGE SO AS TO PREVENT
*        THE POSSIBILITY OF OVERFLOW.
*       NO EXTERNAL RAM IS USED.
*       THE MAGNITUDE OF THE FFT IS COMPUTED AND NORMALIZED SO THAT ITS MAXIMUM
*       VALUE HAS A NONZERO BIT AFTER THE BINARY POINT.
*
******************************************************************************
*
* N IS THE SIZE OF THE TRANSFORM.  N = 2**M.
*
N EQU 128
M EQU 7
*
XIN     EQU 512         * LOCATION OF REAL INPUT DATA
XOUT    EQU 640         * LOCATION OF REAL OUTPUT DATA
XFFT    EQU 768         * LOCATION OF COMPLEX DATA FOR THE FFT
*
* BLOCK B2 DATA MEMORY ALLOCATION (DP = 0 WILL ALWAYS POINT TO B2).
*
XT      EQU 96          * TEMPORARY - REAL PART
YT      EQU 97          * TEMPORARY - IMAGINARY PART
I       EQU 98          * 1ST INDEX
IA      EQU 99          * INDEX TO TWIDDLE FACTORS
IE      EQU 100         * INCREMENT TO IA
HOLDN   EQU 101         * CONTAINS VALUE N
QUARTN  EQU 102         * CONTAINS VALUE N/4
N1      EQU 103         * INCREMENT TO I.
N2      EQU 104         * SEPARATION OF I AND L
J       EQU 105         * LOOP COUNTER
K       EQU 106         * BIT REVERSAL INDEX COUNTER
ONE     EQU 107         * CONTAINS VALUE 1
IADDR   EQU 108         * CONTAINS INPUT
SINTBL  EQU 109         * SINE TABLE POINTER
SIN     EQU 110         * SINE LOCATION
COS     EQU 111         * COSINE LOCATION
MAX     EQU 112         * MAX VALUE OF THE SQR MAGNITUDE OF FFT
*
*
* BEGIN PROGRAM MEMORY SECTION.
*
******************************************************************************
*
        AORG 0
RSVECT  B    32
        AORG 32
INIT    LDPK 0                  * ALWAYS POINT TO B2 FOR TEMP STORAGE
        CNFD
        SOVM
        SSXM                    * 32010 ARITHMETIC
        SPM  1                  * SHIFT PRODUCT LEFT BY 1
        LACK 1
        SACL ONE
        SACL IE                 * INITIALIZE IE = 1
        LALK N
        SACL HOLDN              * HOLDN = N
        SACL N2                 * INITIALIZE N2 = N
        LAC  HOLDN,14
        SACH QUARTN             * QUARTN = N/4
        LRLK AR3,XFFT           * ADDRESS OF COMPLEX INPUT DATA
        SAR  AR3,IADDR          * STORE ON PAGE 0
        LARP 2
        LARK AR1,M-1            * AR1 CONTAINS K COUNTER
*
*   READ IN 128 REAL POINTS
*
        LRLK    AR2,XIN
        RPTK    127
        IN      *+,PA0
*
*   INITIALIZE FFT RAM
*
        LRLK    AR2,XFFT
        ZAC
        SACL    MAX
        RPTK    255
        SACL    *+
*
*   MOVE REAL DATA FROM INPUT LOCATIONS TO COMPLEX FFT LOCATIONS
*
        LRLK    AR2,XFFT
        LARK    AR0,2
        RPTK    127
        BLKD    XIN,*0+
*
*   MOVE SQUARED MAGNITUDE OF PREVIOUS COMPUTATION TO OUTPUT LOCATIONS
*
        LRLK    AR2,XOUT
        RPTK    127
        BLKD    XFFT,*+
*
*   FFT COMPUTATION
*
KLOOP       LAC  N2,15
            SACH N1,1           * N1 = N2
            SACH N2             * N2 = N2/2
            ZAC
            SACL IA             * IA = 0
            SACL J              * J = 0
            LAR  AR2,N2         * AR2 CONTAINS J VALUE
            MAR  *-,AR3         * START AT N2-1
*
           LALK    SINE
           SACL    SINTBL       * SINE TABLE BASE ADDRESS
*
JLOOP          LAC  J,1
               SACL I           * I = J  (DATA ORGANIZED AS REAL VALUE FOLLOWED
*                                 BY IMAGINARY SO THAT ADDRESS I IS 2 TIMES J).
*
ILOOP             LAR  AR0,I
                  LAR  AR3,IADDR        * LOAD INPUT BASE ADDRESS
                  MAR  *0+              * AR3 = I + IADDR
                  LAR  AR0,N1           * ADD N2*2 (N1 = N2*2)
                  LAC  *0+,15           * LOAD (1/2)XI, POINT TO XL
*                                         (L = I + N2)
                  SUB  *,15             * XT = (1/2)(XI - XL)
                  SACH XT               * STORE XT ON PAGE 0
                  ADDH *0-              * XI = (1/2)(XI + XL), POINT TO XI
                  SACH *+               * STORE XI, POINT TO YI
```

```
              LAC  *0+,15          * LOAD (1/2)YI, POINT TO YL
              SUB  *,15            * YT = (1/2)(YI - YL)
              SACH YT
              ADDH *0-             * YI = (1/2)(YI + YL), POINT TO YI
              SACH *0+             * STORE YI, POINT TO YL
*
          LAC     SINTBL
          TBLR    SIN              * READ IN SINE
          ADD     QUARTN
          TBLR    COS              * READ IN COSINE
*
          LT      COS
          MPY     YT               * YT*COS-->P
          LTP     SIN
          MPY     XT               * XT*SIN-->P
          SPAC
          SACH    *-               * YL = XT*COS - XT*SIN
          MPY     YT               * YT*SIN-->P
          LTP     COS
          MPY     XT               * XT*COS-->P
          APAC
          SACH    *                * XL = XT*COS + YT*SIN
*
*
* ADD INCREMENT FOR NEXT LOOP.
*
              LAC  I
              ADD  N1,1            * I = I + N1
              SACL I
              SUB  HOLDN,1         * WHILE I < N
              BLZ  ILOOP
*
          LAC  SINTBL          * INDEX SINTBL POINTER BY IE
          ADD  IE
          SACL SINTBL
*
          LAC  J
          ADD  ONE                 * J = J + 1
          SACL J
          LARP 2
          BANZ JLOOP,*-,AR3
*
       LAC  IE,1
       SACL IE                     * IE = 2 * IE
       LARP 1
       BANZ KLOOP,*-,AR2
*
* DIGIT REVERSE COUNTER FOR RADIX-2 FFT COMPUTATION.
*
DRC2   ZAC
       SACL I
       SACL J
       LAR  AR0,IADDR
       LARP 4
       LAR  AR4,HOLDN              * FOR I = 0 TO N-2
       MAR  *-
       MAR  *-,AR2
DRLOOP        SUB  J               * IF I < J, THEN SWAP
              BGEZ NOSWAP
*
                     LAR  AR2,J
                     MAR  *0+,AR3  * J = J + IADDR
                     LAR  AR3,I
                     MAR  *0+,AR2  * I = I + IADDR
* SWAP I AND L VALUES.
                     ZALH *,AR3
                     ADDS *
                     SACH *+,0,AR2 * X(I) = X(J)
                     SACL *+       * X(J) = X(I)
*
                     ZALH *,AR3
                     ADDS *
                     SACH *,0,AR2  * Y(I) = Y(J)
                     SACL *,0,AR3  * Y(J) = Y(I)
NOSWAP          LAC  HOLDN
                SACL K                         * K = N
INLOOP               LAC  J
                     SUB  K                    * IF J >= K THEN
                     BLZ  OUTLOP,*,AR4
                          SACL J               * J = J - K
                          LAC  K,15
                          SACH K               * K = K/2.
                          B    INLOOP
OUTLOP          ADD  K,1
                SACL J                         * L = L + J
                LAC  I
                ADD  ONE,1
                SACL I                         * INCREMENT I
                BANZ DRLOOP,*-,AR2
*
*
*   COMPUTE THE MAXIMUM VALUE OF THE SQUARED MAGNITUDE OF FFT
*
        LRLK    AR2,XFFT
        LARK    AR3,127
*
LOOP1   ZAC
        MPYK    0
        SQRA    *+                        * X(I)**2
        SQRA    *+,AR3                    * Y(I)**2
        APAC
        SACH    IADDR                     * IADDR = X(I)**2 + Y(I)**2
        SUBH    MAX
        BLEZ    CONT
        LAC     IADDR
        SACL    MAX
CONT    BANZ    LOOP1,*-,AR2
*
*   NORMALIZE THE MAX VALUE TO FIND EXPONENT
*
        LARK    AR2,0
        ZALH    MAX
        RPTK    14
        NORM                              * AR2 CONTAINS EXPONENT
        SAR     AR2,I                     * STORE EXPONENT IN I
*
*   COMPUTE SQUARED MAGNITUDE OF FFT
*
        LRLK    AR2,XFFT
        LRLK    AR1,XFFT
        LARK    AR3,127
*
LOOP2   ZAC
        MPYK    0
        SQRA    *+                        * X(I)**2
        SQRA    *+,AR1                    * Y(I)**2
```

```
        APAC
        SFR
        RPT     I
        SFL                               * NORMALIZE RESULT
        SACH    *+,0,AR3                  * XOUT(I) = X(I)**2 + Y(I)**2
        BANZ    LOOP2,*-,AR2
*
*    OUTPUT SQUARED MAGNITUDE
*
        LRLK    AR2,XFFT
        RPTK    127
        OUT     *+,PA2
*
*  FFT COMPLETE.
*
WHOA    B WHOA
*
* COEFFICIENT TABLE (SIZE OF TABLE IS 3N/4).
*
SINE EQU $
      DATA >0
      DATA >648
      DATA >C8C
      DATA >12C8
      DATA >18F9
      DATA >1F1A
      DATA >2528
      DATA >2B1F
      DATA >30FC
      DATA >36BA
      DATA >3C57
      DATA >41CE
      DATA >471D
      DATA >4C40
      DATA >5134
      DATA >55F6
      DATA >5A82
      DATA >5ED7
      DATA >62F2
      DATA >66D0
      DATA >6A6E
      DATA >6DCA
      DATA >70E3
      DATA >73B6
      DATA >7642
      DATA >7885
      DATA >7A7D
      DATA >7C2A
      DATA >7D8A
      DATA >7E9D
      DATA >7F62
      DATA >7FD9
COSINE EQU $
      DATA >7FFF
      DATA >7FD9
      DATA >7F62
      DATA >7E9D
      DATA >7D8A
      DATA >7C2A
      DATA >7A7D
      DATA >7885
      DATA >7642
      DATA >73B6
      DATA >70E3
      DATA >6DCA
      DATA >6A6E
      DATA >66D0
      DATA >62F2
      DATA >5ED7
      DATA >5A82
      DATA >55F6
      DATA >5134
      DATA >4C40
      DATA >471D
      DATA >41CE
      DATA >3C57
      DATA >36BA
      DATA >30FC
      DATA >2B1F
      DATA >2528
      DATA >1F1A
      DATA >18F9
      DATA >12C8
      DATA >C8C
      DATA >648
      DATA >0
      DATA >F9B8
      DATA >F374
      DATA >ED38
      DATA >E707
      DATA >E0E6
      DATA >DAD8
      DATA >D4E1
      DATA >CF04
      DATA >C946
      DATA >C3A9
      DATA >BE32
      DATA >B8E3
      DATA >B3C0
      DATA >AECC
      DATA >AA0A
      DATA >A57E
      DATA >A129
      DATA >9D0E
      DATA >9930
      DATA >9592
      DATA >9236
      DATA >8F1D
      DATA >8C4A
      DATA >89BE
      DATA >877B
      DATA >8583
      DATA >83D6
      DATA >8276
      DATA >8163
      DATA >809E
      DATA >8027
      END
```

APPENDIX G

A 256-POINT, RADIX-2 DIF FFT IMPLEMENTATION (LOOPED CODED)

```
        IDT  'FFT2'
*
*    COOLEY-TUKEY 256-POINT, RADIX-2, DIF FFT PROGRAM FOR THE TMS32020.
*
*       SINGLE FFT BUTTERFLY.
*       COMPLEX INPUT DATA, STORED AS X(I), Y(I), X(I+1), Y(I+1), ...
*       USES TABLE LOOKUP (FROM EXTERNAL DATA MEMORY) OF THE TWIDDLE FACTORS.
*       INTERMEDIATE VALUES ARE SCALED BY .5 AT EACH STAGE SO AS TO PREVENT
*        THE POSSIBILITY OF OVERFLOW.
*
**************************************************************************
*
* N IS THE SIZE OF THE TRANSFORM.  N = 2**M.
*
N EQU 256
M EQU 8
*
INPUT   EQU 512         * LOCATION OF COMPLEX INPUT DATA IN INTERNAL DATA MEM
TABLE   EQU 1024        * LOCATION OF COEFFICIENT TABLE IN EXTERNAL DATA MEM
*
* BLOCK B2 DATA MEMORY ALLOCATION (DP = 0 WILL ALWAYS POINT TO B2).
*
XT      EQU 96          * TEMPORARY - REAL PART
YT      EQU 97          * TEMPORARY - IMAGINARY PART
I       EQU 98          * 1ST INDEX
IA      EQU 99          * INDEX TO TWIDDLE FACTORS
IE      EQU 100         * INCREMENT TO IA
HOLDN   EQU 101         * CONTAINS VALUE N
QUARTN  EQU 102         * CONTAINS VALUE N/4
N1      EQU 103         * INCREMENT TO I.
N2      EQU 104         * SEPARATION OF I AND L
J       EQU 105         * LOOP COUNTER
K       EQU 106         * BIT REVERSAL INDEX COUNTER
ONE     EQU 107         * CONTAINS VALUE 1
IADDR   EQU 108         * CONTAINS INPUT
COSTBL  EQU 109         * COSTBL = TABLE + N/4
*
*
* BEGIN PROGRAM MEMORY SECTION.
*
**************************************************************************
*
        AORG 0
RSVECT  B    32
        AORG 32
INIT    LDPK 0                  * ALWAYS POINT TO B2 FOR TEMP STORAGE
        SOVM
        SSXM                    * 32010 ARITHMETIC
        SPM  1                  * SHIFT PRODUCT LEFT BY 1
        LACK 1
        SACL ONE
        SACL IE                 * INITIALIZE IE = 1
        LALK N
        SACL HOLDN              * HOLDN = N
        SACL N2                 * INITIALIZE N2 = N
        LAC  HOLDN,14
        SACH QUARTN             * QUARTN = N/4
        LRLK AR3,INPUT          * ADDRESS OF COMPLEX INPUT DATA
        SAR  AR3,IADDR          * STORE ON PAGE 0
        LRLK AR4,TABLE          * ADDRESS OF SINE TABLE
        LARP 4
        RPTK 191                * MOVE 192 COEFFICIENTS
        BLKP SINE,*+
        LRLK AR4,TABLE          * ADDRESS OF SINE TABLE
        LAR  AR0,QUARTN
        MAR  *0+,AR2            * COSTBL = TABLE + N/4, point to J counter
        SAR  AR4,COSTBL
        LARK AR1,M-1            * AR1 CONTAINS K COUNTER
*
*   READ IN 256 COMPLEX POINTS
*
        LRLK    AR2,INPUT
        RPTK    255
        IN      *+,PA0
        RPTK    255
        IN      *+,PA0
*
*   FFT COMPUTATION
*
KLOOP      LAC  N2,15
           SACH N1,1            * N1 = N2
           SACH N2              * N2 = N2/2
           ZAC
           SACL IA              * IA = 0
           SACL J               * J = 0
           LAR  AR2,N2          * AR2 CONTAINS J VALUE
           MAR  *-,AR3          * START AT N2-1
*
           LAR  AR4,COSTBL      * COSINE TABLE BASE ADDRESS
*
JLOOP         LAC  J,1
              SACL I            * I = J  (DATA ORGANIZED AS REAL VALUE FOLLOWED
*                                 BY IMAGINARY SO THAT ADDRESS I IS 2 TIMES J).
*
ILOOP            LAR  AR0,I
                 LAR  AR3,IADDR        * LOAD INPUT BASE ADDRESS
                 MAR  *0+              * AR3 = I + IADDR
                 LAR  AR0,N1           * ADD N2*2 (N1 = N2*2)
                 LAC  *0+,15           * LOAD (1/2)XI, POINT TO XL
*                                        (L = I + N2)
                 SUB  *,15             * XT = (1/2)(XI - XL)
                 SACH XT               * STORE XT ON PAGE 0
                 ADDH *0-              * XI = (1/2)(XI + XL), POINT TO XI
                 SACH *+               * STORE XI, POINT TO YI
                 LAC  *0+,15           * LOAD (1/2)YI, POINT TO YL
                 SUB  *,15             * YT = (1/2)(YI - YL)
                 SACH YT
                 ADDH *0-              * YI = (1/2)(YI + YL), POINT TO YI
                 SACH *0+,0,AR4        * STORE YI, POINT TO YL
*                                        AR4 POINTS TO COSTBL
                 LAR  AR0,QUARTN
                 LT   *0-              * LOAD T WITH COS, POINT TO SIN
                 MPY  YT
                 LTP  *0+,AR3          * ACC <-- C*YT, POINT TO YL
                 MPY  XT
                 SPAC                  * YL = C*YT - S*XT
                 SACH *-,0,AR4         * STORE YL
                 MPY  YT
                 LTP  *,AR3            * ACC <-- S*YT, POINT TO XL
                 MPY  XT
                 APAC                  * XL = C*XT + S*YT
                 SACH *                * STORE XL
*
*
* ADD INCREMENT FOR NEXT LOOP.
```

```
*
                    LAC  I
                    ADD  N1,1               * I = I + N1
                    SACL I
                    SUB  HOLDN,1            * WHILE I < N
                    BLZ  ILOOP
*
                LAR  AR0,IE             * INDEX COSTBL POINTER BY IE
                LARP 4
                MAR  *0+,AR2
                LAC  J
                ADD  ONE                    * J = J + 1
                SACL J
                BANZ JLOOP,*-,AR3
*
             LAC  IE,1
             SACL IE                        * IE = 2 * IE
             LARP 1
             BANZ KLOOP,*-,AR2
*
* DIGIT REVERSE COUNTER FOR RADIX-2 FFT COMPUTATION.
*
DRC2    ZAC
        SACL I
        SACL J
        LAR  AR0,IADDR
        LARP 4
        LAR  AR4,HOLDN                      * FOR I = 0 TO N-2
        MAR  *-
        MAR  *-,AR2
DRLOOP          SUB  J                      * IF I < J, THEN SWAP
                BGEZ NOSWAP
*
                        LAR  AR2,J
                        MAR  *0+,AR3    * J = J + IADDR
                        LAR  AR3,I
                        MAR  *0+,AR2    * I = I + IADDR
* SWAP I AND L VALUES.
                        ZALH *,AR3
                        ADDS *
                        SACH *+,0,AR2   * X(I) = X(J)
                        SACL *+         * X(J) = X(I)
*
                        ZALH *,AR3
                        ADDS *
                        SACH *,0,AR2    * Y(I) = Y(J)
                        SACL *,0,AR3    * Y(J) = Y(I)
NOSWAP          LAC  HOLDN
                SACL K                              * K = N
INLOOP                  LAC  J
                        SUB  K                      * IF J >= K THEN
                        BLZ  OUTLOP,*,AR4
                                SACL J              * J = J - K
                                LAC  K,15
                                SACH K              * K = K/2.
                                B    INLOOP
OUTLOP          ADD  K,1
                SACL J                              * L = L + J
                LAC  I
                ADD  ONE,1
                SACL I                              * INCREMENT I
                BANZ DRLOOP,*-,AR2
*
```

```
*   OUTPUT FFT VALUES
*
        LRLK    AR2,INPUT
        RPTK    255
        OUT     *+,PA1
        RPTK    255
        OUT     *+,PA1
*
*  FFT COMPLETE.
*
WHOA    B WHOA
*
* COEFFICIENT TABLE (SIZE OF TABLE IS 3N/4).
*
SINE EQU $
      DATA >0
      DATA >324
      DATA >648
      DATA >96B
      DATA >C8C
      DATA >FAB
      DATA >12C8
      DATA >15E2
      DATA >18F9
      DATA >1C0C
      DATA >1F1A
      DATA >2224
      DATA >2528
      DATA >2827
      DATA >2B1F
      DATA >2E11
      DATA >30FC
      DATA >33DF
      DATA >36BA
      DATA >398D
      DATA >3C57
      DATA >3F17
      DATA >41CE
      DATA >447B
      DATA >471D
      DATA >49B4
      DATA >4C40
      DATA >4EC0
      DATA >5134
      DATA >539B
      DATA >55F6
      DATA >5843
      DATA >5A82
      DATA >5CB4
      DATA >5ED7
      DATA >60EC
      DATA >62F2
      DATA >64E9
      DATA >66D0
      DATA >68A7
      DATA >6A6E
      DATA >6C24
      DATA >6DCA
      DATA >6F5F
      DATA >70E3
      DATA >7255
      DATA >73B6
      DATA >7505
```

```
       DATA >7642
       DATA >776C
       DATA >7885
       DATA >798A
       DATA >7A7D
       DATA >7B5D
       DATA >7C2A
       DATA >7CE4
       DATA >7D8A
       DATA >7E1E
       DATA >7E9D
       DATA >7F0A
       DATA >7F62
       DATA >7FA7
       DATA >7FD9
       DATA >7FF6
COSINE EQU $
       DATA >7FFF
       DATA >7FF6
       DATA >7FD9
       DATA >7FA7
       DATA >7F62
       DATA >7F0A
       DATA >7E9D
       DATA >7E1E
       DATA >7D8A
       DATA >7CE4
       DATA >7C2A
       DATA >7B5D
       DATA >7A7D
       DATA >798A
       DATA >7885
       DATA >776C
       DATA >7642
       DATA >7505
       DATA >73B6
       DATA >7255
       DATA >70E3
       DATA >6F5F
       DATA >6DCA
       DATA >6C24
       DATA >6A6E
       DATA >68A7
       DATA >66D0
       DATA >64E9
       DATA >62F2
       DATA >60EC
       DATA >5ED7
       DATA >5CB4
       DATA >5A82
       DATA >5843
       DATA >55F6
       DATA >539B
       DATA >5134
       DATA >4EC0
       DATA >4C40
       DATA >49B4
       DATA >471D
       DATA >447B
       DATA >41CE
       DATA >3F17
       DATA >3C57
       DATA >398D
       DATA >36BA
       DATA >33DF
       DATA >30FC
       DATA >2E11
       DATA >2B1F
       DATA >2827
       DATA >2528
       DATA >2224
       DATA >1F1A
       DATA >1C0C
       DATA >18F9
       DATA >15E2
       DATA >12C8
       DATA >FAB
       DATA >C8C
       DATA >96B
       DATA >648
       DATA >324
       DATA >0
       DATA >FCDC
       DATA >F9B8
       DATA >F695
       DATA >F374
       DATA >F055
       DATA >ED38
       DATA >EA1E
       DATA >E707
       DATA >E3F4
       DATA >E0E6
       DATA >DDDC
       DATA >DAD8
       DATA >D7D9
       DATA >D4E1
       DATA >D1EF
       DATA >CF04
       DATA >CC21
       DATA >C946
       DATA >C673
       DATA >C3A9
       DATA >C0E9
       DATA >BE32
       DATA >BB85
       DATA >B8E3
       DATA >B64C
       DATA >B3C0
       DATA >B140
       DATA >AECC
       DATA >AC65
       DATA >AA0A
       DATA >A7BD
       DATA >A57E
       DATA >A34C
       DATA >A129
       DATA >9F14
       DATA >9D0E
       DATA >9B17
       DATA >9930
       DATA >9759
       DATA >9592
       DATA >93DC
       DATA >9236
       DATA >90A1
       DATA >8F1D
       DATA >8DAB
       DATA >8C4A
       DATA >8AFB
       DATA >89BE
       DATA >8894
       DATA >877B
       DATA >8676
       DATA >8583
       DATA >84A3
       DATA >83D6
       DATA >831C
       DATA >8276
       DATA >81E2
       DATA >8163
       DATA >80F6
       DATA >809E
       DATA >8059
       DATA >8027
       DATA >800A
       END
```