An Implementation of FFT, DCT, and Other Transforms on the TMS320C30

APPLICATION REPORT: SPRA113

Panos Papamichalis Regional Technology Center Waltham, Massachusetts Texas Instruments

Digital Signal Processing Solutions



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.

Copyright © 1997, Texas Instruments Incorporated

TRADEMARKS

TI is a trademark of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.

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

An Implementation of FFT, DCT, and Other Transforms on the TMS320C30

Abstract

This book describes the several types of transforms and related algorithms used on the TMS320C30 family of digital signal processors. These include:

- □ The Fast Fourier Transforms (FFTs)
 - the complex radix-2 FFT
 - the complex radix-4 FFT
 - the real valued radix-2
- ☐ The Discrete Hartley Transform (DHT)
- □ The Discrete Cosine Transform (DCT)

The book contains:

- ☐ A description of transforms and their implementation on the TMS32030 family of digital signal processors.
- □ A description and comparison of the different kinds of transforms: the FFTs, the Hartley transform and the Cosine transform
- ☐ A description of the features of the TMS320C30 that allow the efficient implementation of these algorithms
- □ Outlines of specific descriptions of implementations, transforms and TMS320C30 C Compiler facts
- Implementation issues
- Several graphics and tables detailing
 - Forms and flowgraphs of FFTs



- Memory requirements for FFT and Hartley transforms
- Differences in FFT and DCT timing

The end of the book contains 17 appendices with actual TMS320C30 source code for performing transforms.



Product Support

World Wide Web

Our World Wide Web site at www.ti.com contains the most up to date product information, revisions, and additions. New users must register with TI&ME before they can access the data sheet archive. TI&ME allows users to build custom information pages and receive new product updates automatically via email.

Email

For technical issues or clarification on switching products, please send a detailed email to *dsph@ti.com*. Questions receive prompt attention and are usually answered within one business day.

This report describes the implementation of several Fast Fourier Transforms (FFTs) and related algorithms on the TMS320C30. The TMS320C30 is the first device in the third generation of 32-bit floating-point Digital Signal Processors (DSPs) in the Texas Instruments TMS320 family. The algorithms considered here are the complex radix-2 FFT, the complex radix-4 FFT, the real-valued radix-2 FFT (both forward and inverse transforms), the Discrete Hartley Transform (DHT), and the Discrete Cosine Transform (DCT). These transforms have many applications, such as in image processing, sonar, and radar.

The introduction briefly describes transforms and their implementation on the TMS320 family of processors. Next, the different kinds of FFTs (including the real FFT), the closely-related Hartley transform, and the Cosine transform are described and compared. This is followed by a description of the TMS320C30 features that permit efficient implementations of these algorithms. Then, specific implementations, transforms, and TMS320C30 C Compiler facts are outlined. Finally, the report discusses some implementation issues, and the appendices list actual TMS320C30 code for performing transforms.

The powerful architecture and instruction set of the TMS320C30 permit flexible and compact coding of the algorithms in assembly language while preserving close correspondence to a high-level language implementation. The efficiency of the architecture and the speed of the device make faster realization of real and complex transforms possible. With the availability of a C compiler, these routines can be put in C-callable form and used as faster versions of FFT C functions.

Introduction

The Fast Fourier Transform (FFT) is an important tool used in Digital Signal Processing (DSP) applications. Its development by Cooley and Tuckey gave impetus to the establishment of DSP as an independent discipline. The well-structured form of the FFT has also made it one of the benchmarks in assessing the performance of number-crunching devices and systems.

In recent years, because of the popularity of this signal-processing tool, there have been efforts to improve its performance by advances both at the algorithmic level and in hardware implementation. Researchers have been developing efficient algorithms to increase the execution speed of FFTs while keeping requirements for memory size low. On the other hand, developers of VLSI systems are including features in their designs that improve system performance for applications requiring FFTs. In particular, single-chip programmable DSP devices, currently available or under development, can realize FFTs with speeds that allow the implementation of very complex systems in realtime.

The Texas Instruments TMS320 family consists of five generations of programmable digital signal processors. The TMS32010 introduced the first generation, which today encompasses more than twelve devices with various speeds, interfacing capabilities, and price/performance combinations. FFT implementations on the TMS32010 can be found in the appendix of the book by Burrus and Parks [1].

The second-generation TMS320 devices (the TMS32020, the TMS320C25, and their spinoffs) enhanced the architecture and speed capabilities of the first generation. Examples of FFT programs implemented on the TMS32020 can be found in an application report in the book *Digital Signal Processing Applications with the TMS320 Family* [2]. Such programs are easily extended to the TMS320C25 because of the code compatibility between devices.

The architectural and speed improvements on the processors from one generation to the next have made the FFT computation faster and the programming easier. These advantages have reached a new high level in the third generation. The TMS320C30 is the first device in the third generation, and this report examines implementation of the FFT algorithms on it. The fourth generation (TMS320C4x) is a new set of floating-point devices, while the fifth generation (TMS320C5x) is a continuation of the fixed-point devices. Since software compatibility is maintained within the fixed-point and the floating-point devices, the existing FFT implementations will also be applicable to these new generations.

The Fourier Transform of an analog signal x(t), given as

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$$
 (1)

determines the frequency content of the signal x(t). In other words, for every frequency, the Fourier transform $X(\omega)$ determines the contribution of a sinusoid of that frequency in the composition of the signal x(t). For computations on a digital computer, the signal x(t) is sampled at discrete-time instants. If the input signal is digitized, a sequence of numbers x(n) is available instead of the continuous-time signal x(t). Then, the Fourier transform takes the form

$$X(e^{j\omega}) = \sum_{n = -\infty}^{\infty} x(n) e^{-j\omega n}$$
(2)

The resulting transform $X(e^{j\omega})$ is a periodic function of ω , and it needs to be computed for only one period. The actual computation of the Fourier transform of a stream of data presents difficulties because $X(e^{j\omega})$ is a continuous function in ω . Since the transform must be computed at discrete points, the properties of the Fourier transform led to the definition of the *Discrete Fourier Transform* (DFT), given by

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi kn}{N}}$$
 (3)

When x(n) consists of N points x(0), x(1), . . . , x(N-1), the frequency-domain representation is given by the set of N points X(k), $k=0,1,\ldots,N-1$. Equation (3) is often written in the form

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$
(4)

where W $_N^{nk} = e - j \, 2\pi nk \, / \, N$. The factor W_N is sometimes referred to as the *twiddle factor*. A detailed description of the DFT can be found in references [1,3,4]. The computational requirements of the DFT increase rapidly with increasing block size N, having an impact on the real-time system performance. This problem was alleviated with the development of special fast algorithms, collectively known as Fast Fourier Transform (FFT). With an FFT, the computational burden increases much less rapidly with N, and for any given N, the FFT computational load, measured in terms of required multiplications and additions, is smaller than a brute-force computation of the DFT.

The definition of the FFT is identical to the DFT: only the method of computation differs. To achieve the efficiency of an FFT, it is important that N be a highly composite number. Typically, the length N of the FFT is a power of 2: $N = 2^M$, and the whole algorithm breaks down into a repeated application of an elementary transform known as a butterfly. If N is not a power of 2, the sequence x(n) is appended with enough zeroes to make the total length a power of 2. Again, references [1,3,4] contain a detailed development of the FFT. Reference [2] also discusses the same topic.

Different Forms of the FFT

Over the years, researchers have developed different forms of FFT for more efficient computation. Special cases, such as those in which the input is a sequence of real numbers, have been investigated, and even more sophisticated algorithms have been developed. The general form of the FFT butterfly is given in Figure 1.

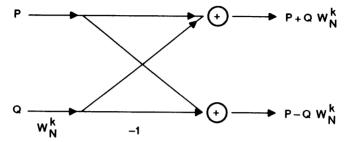


Figure 1. Radix-2 Butterfly for Decimation in Time

If the inputs to the butterfly are the two complex numbers P and Q, the outputs will be the complex numbers P' and Q', such that

$$P' = P + Q W_N^k \tag{5}$$

and

$$Q' = P - Q W_N^k \tag{6}$$

The quantities P, Q, and P', Q' represent different points in the array being transformed, and they may or may not occupy adjacent locations in that array. In an in-place computation, the result P' will overwrite P, and Q' will overwrite Q. W_N^k represents again

the twiddle factor, and its exponent is determined by the location of the corresponding butterfly in the FFT algorithm.

Figure 2 shows an alternate form of the same FFT butterfly.

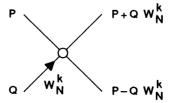


Figure 2. Alternate Form of Radix-2 Butterfly for Decimation in Time.

Although the notation is now less descriptive, it creates a clearer picture when several butterflies are put together to form an FFT. Using the first notation, Figure 3 is the flowgraph of an 8-point FFT example.

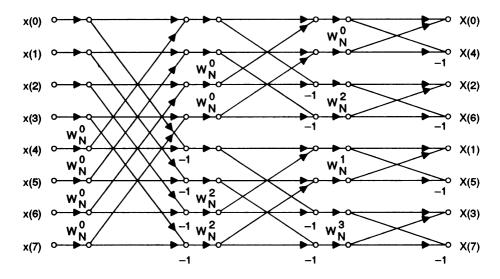


Figure 3. Example of 8-Point FFT with Decimation in Time.

Note that the input sequence x(n) is in the correct order, while the output X(k) is scrambled. Actually, this scrambling occurs in a very systematic way, called bit-reversed order: If you express the indices of a scrambled sequence in binary and you reverse this number, the result is the order that this particular point occupies. For instance, X(3) occupies the sixth position in the output (when counting from the zero position). In binary form, $3_{10} = 011_2$, and if bit-reversed, you get $110_2 = 6_{10}$, which is the position that X(3) occupies. It turns out that the third position is occupied by X(6), and to restore the correct order at the output, you need only to swap these two numbers.

The same procedure can be repeated with all the scrambled numbers not occupying the position that their index suggests. If the input sequence x(n) is rearranged to appear in bit-reversed form, the output X(k) appears in the correct order, as shown in Figure 4.

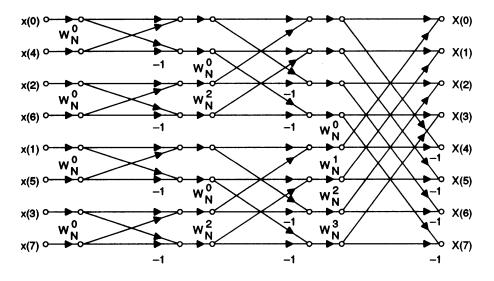


Figure 4. Alternate Form of 8-Point FFT with Decimation in Time. The Input Is in Bit-Reversed Order and the Output Is in the Correct Order.

Since the only difference between Figures 3 and 4 is a rearrangement of the butterflies, the computational load and the final results are identical. In terms of implementation, this rearrangement means that the nesting of the two innermost loops in the FFT routine is interchanged.

The butterflies and the FFT configurations presented thus far implement the FFT with a decimation in time. This terminology essentially describes a way of grouping the terms of the DFT definition; see Equation (3). An alternative way of grouping the DFT terms together is called decimation in frequency. Figures 5 and 6 show the same example of an 8-point FFT: Figure 5 with the input in correct order and the output in bit-reversed order, and Figure 6 vice-versa, and using the decimation in frequency (DIF).

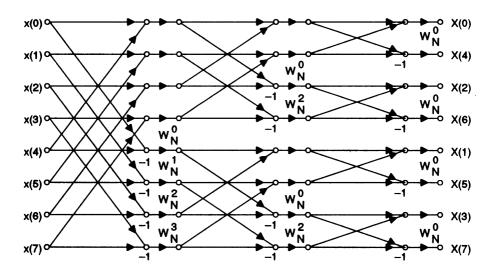


Figure 5. Example of an 8-Point FFT with Decimation in Frequency.

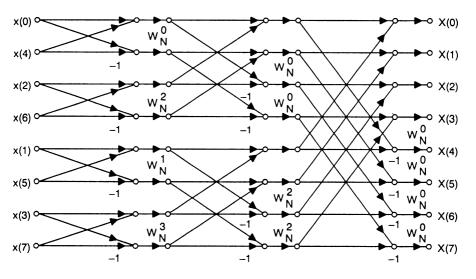


Figure 6. Alternate Form of 8-Point FFT with Decimation in Frequency. The Input Is in Bit-Reversed Order and the Output Is in the Correct Order

Pictorially, the difference between decimation in time and decimation in frequency is that the twiddle factor appears at the input of the butterfly in the first, and at the output in the second. Otherwise, the two methods are identical in terms of results. However, depending on what is the most convenient order of getting the twiddle factors and where the longest-span butterfly appears, you may prefer one method over the other.

The butterfly shown in Figure 1 (or Figure 2) is the smallest element in a radix-2 FFT. The radix of the FFT represents the number of inputs that are combined in a butterfly. The Fast Fourier Transform is usually explained around the radix-2 algorithm for conceptual simplicity. If, however, higher-order radices are used, more computational savings can be achieved. These savings increase with the radix, but there is very little improvement above radix 4. That's why the radix-2 and radix-4 FFTs are the most commonly used algorithms.

In radix-4 FFT, each butterfly has 4 inputs and 4 outputs, essentially combining two stages of a radix-2 algorithm in one. Figure 7 shows this combination graphically.

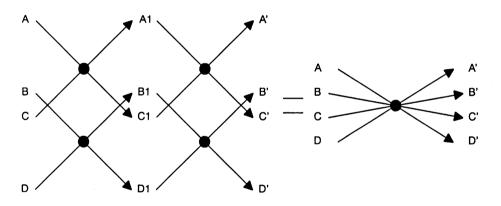


Figure 7. Butterfly for Radix-4, Decimation-in-Time FFT.

Although four radix-2 butterflies are combined into one radix-4 butterfly, the computational load of the latter is less than four times the load of a radix-2 butterfly. Examples of radix-4, 16-point FFTs are shown in Figures 8 and 9 for decimation in time and decimation in frequency, respectively.

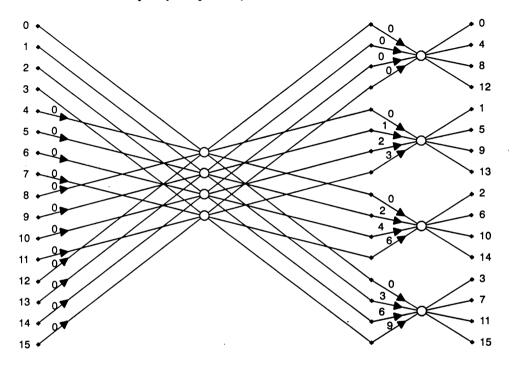


Figure 8. Example of a 16-Point, Radix-4, Decimation-in-Time FFT.

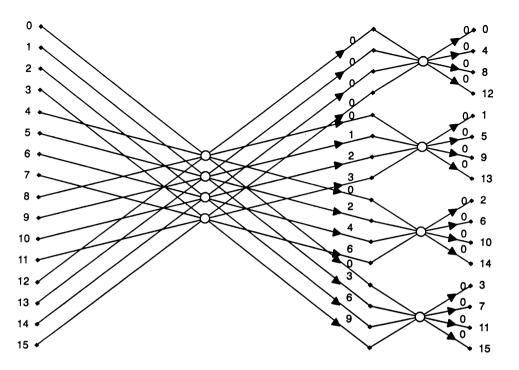


Figure 9. Example of a 16-Point, Radix-4, Decimation-in-Frequency FFT.

These configurations take the incoming sequence in order and produce the frequency-domain result in digit-reversed form. It is a simple matter to rearrange the FFT and have the input in digit-reversed form and the output in order.

Digit reversal is similar to bit reversal, except that the number whose digits are reversed is written in base 4 (equal to the radix) rather than base 2. For example, the output value X(14) in a 16-point, radix-4 FFT occupies position eleven (again starting from zero) because $14_{10} = 32_4$ and, reversing the digits of the number, $23_4 = 11_{10}$. To restore the output to the correct order, the contents of locations with digit-reversed indices should be swapped. However, since the TMS320C30 has a special bit-reversed addressing mode, it is desirable to have the output of the radix-4 computation in bit-reversed rather than digit-reversed form. This is accomplished quite simply if, in each radix-4 butterfly, the two middle output legs are interchanged. That is, whenever the output of the butterfly is the four numbers A', B', C', and D', instead of storing them in that order, store them in the order A', C', B', and D', as shown in Figure 10.

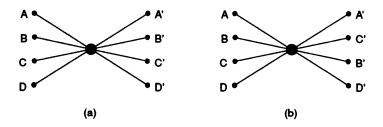


Figure 10. Radix-4 Butterflies. (a) Regularly-Ordered Output, (b) Bit-Reversed Output.

References [5, 6] explain why this simple rearrangement puts the result in bit-reversed order.

Features of the TMS320C30

The TMS320C30 is the first device introduced in the third generation of the TMS320 Digital Signal Processors [7,8]. It has many architectural features that permit very efficient implementation of algorithms. Some of those features pertinent to the FFT implementation are discussed in this section.

The two most salient characteristics of the TMS320C30 device are its high speed (60-ns cycle time) and floating-point arithmetic. The higher speed makes the implementation of real-time application easier than in earlier processors, even when the other architectural advantages are not considered. Each instruction executes in a single cycle under mild pipeline restrictions. The device automatically takes care of any potential conflicts. The pipeline should be observed closely (e.g., using the trace capability of the simulator) only if code optimization for speed is required.

The floating-point capability permits the handling of numbers of high dynamic range without concern for overflows. In FFT programs, in particular, the computed values tend to increase from one stage to the next, as discussed in reference [2]. Then, the fixed-point arithmetic will cause overflows if the incoming numbers are large enough and no provisions are made for scaling. All these considerations are eliminated with the floating-point capability of the TMS320C30. The TMS320C30 performs floating-point arithmetic with the same speed as any fixed point operation; no performance is sacrificed for this feature.

There are eight extended-precision registers, R0—R7, that can be used as accumulators or general-purpose registers, and eight auxiliary registers, AR0—AR7, for addressing and integer arithmetic. For many applications, these registers are sufficient for temporary storage of values, and there is no need to use memory locations. This is the case with the radix-2 FFT algorithm, where no locations are required other than those for the transformation of incoming data to be transformed. Also, arithmetic using these registers greatly increases the programming efficiency. The two index registers, IR0 and IR1, are used for indexing the contents of the auxiliary registers AR0—AR7, thus making the access of the butterfly legs and the twiddle factors easy.

A powerful structure in the TMS320C30 is the block-repeat capability that has the form

RPTB LABEL
put instructions here
last instruction

LABEL

Whatever occurs after the RPTB instruction and up to the LABEL is repeated one time more than the number included in the repeat counter register, RC. The RC register must be initialized before entering the block-repeat construct. The net effect is that the repeated code behaves as if it were straight-line coded (no penalty for looping), with program size equal to the one in looped code. In this way, the FFT butterfly, being the core of the program, can be implemented in a block-repeat form, thereby saving execution time while preserving the clarity of the program and conserving program space.

A bit-reversed addressing mode is available to eliminate the need for swapping memory locations at the beginning or the end of the FFT (depending on the FFT type). When you use this addressing mode, you access a sequence of data points in bit-reversed order rather than sequentially, and you can recover the points in the correct order during retrieval of the data instead of spending extra cycles to accomplish it in software.

Implementation of Radix-2 and Radix-4 Complex FFTs

Because of the powerful architecture and the instruction set of the TMS320C30, the assembly language program follows closely the flow of a high-level language program; this makes it easy to read and debug. It also keeps the size of the program small and reduces the requirements for program memory. Appendix A presents an example of code for a Radix-2 complex FFT, while Appendix B is a radix-4 complex FFT. The program memory requirements for these programs (as well as others to be discussed later) are given in Table 1.

Table 1. Program Memory Requirements for the Core of the FFT and Hartley
Transforms

Routine Type	Program Size	
Radix-2, complex FFT	50 words	
Radix-4, complex FFT	170 words	
Radix-2, real FFT	68 words	
Radix-2, real inverse FFT	76 words	
Hartley transform	71 words	

The numbers in the table correspond only to the core program and do not include the sine/cosine tables for the twiddle factors, any input/output, or any bit-reversing operations. Note also that they are independent of the FFT data size.

The data memory requirements are, of course, dependent on the FFT size. The maximum length of a complex, radix-2 FFT that can be implemented entirely on the internal memory of the TMS320C30 is 1024 points. In the present implementation, the 1024-point radix-4 FFT requires a few more locations (about 7) than are available on-chip.

The code (provided in the appendices) has been written to be independent of the FFT length. The length N, together with the sine/cosine tables for the twiddle factors, should be provided separately to maintain the generic nature of the core FFT program. An example of a file with the sine/cosine tables for a 64-point FFT is given in the Appendix F. Note that the FFT size and the number of stages are declared .global in both files (i.e., the main routine and the file with the table) so that the core program gets the actual values during linking.

To reduce the storage requirements of a sine/cosine table, a full sine and a cosine cycle are overlapped. The table stores 5/4 of a full sine wave, with the cosine table starting with a phase delay of 1/4 cycle from the sine table. This table size is larger than actually needed, and it is selected merely for testing convenience of the algorithms. The minimum table size for a radix-2 complex FFT includes 1/2 of a full sine wave, and 1/2 of a full cosine wave. If these two half waves are combined using the above quarter-cycle phase delay, the minimum table size for this kind of FFT is 3/4 of a full sine wave. For instance, for a 1024-point FFT, the table can be the first 768 points of a sine wave, where a full cycle would be 1024 points. In the case of a radix-4 complex FFT, the minimum table size should include 3/4 of a sine and 3/4 of a cosine wave. Overlapping these requirements, we get the minimum table size of a radix-4 algorithm to be one full sine wave.

An example of a linking file is also included in Appendix F to show how the different segments are assigned. For a complete description of the assembler and linker, consult the corresponding manual [6].

The timing of the FFT routines was done using the cycle-counting capability of the TMS320C30 simulator. For the conversion of the number of cycles into seconds, a cycle time of 50 ns was used. The timing refers only to the core FFT computation, ignoring read-in and write-out requirements, since such requirements are application-dependent. Also, no bit reversal is counted (although it may be included in the program), since it is performed as part of the read-in or read-out. Table 2 gives the timing for the different FFT routines and for the Hartley transform.

Transform Size	Radix-2 Complex FFT	Radix-4 Complex FFT	Radix-2 Real FFT	Radix-2 Real Inverse FFT	Hartley Transform
64	0.101	0.103	0.047	0.053	0.068
128	0.211	_	0.099	0.110	0.151
256	0.453	0.520	0.215	0.241	0.336
512	0.991	_	0.476	0.535	0.943
1024	2.175	2.533	1.055	1.193	2.025
1024	1.972				

Table 2. FFT Timing in Milliseconds[†]

The last entry in this table represents the timing of the radix-2, DIT routine generated at the University of Erlangen [18] and given in Appendix A. These numbers are typically used for benchmarking.

Implementation of Real FFT

The development of FFT algorithms is centered mostly around the assumption that the input sequence consists of complex numbers (as does the output). This assumption guarantees the generality of the algorithm. However, in a large number of actual applications, the input is a sequence of real numbers. If this condition is taken into consideration, additional computational savings can be achieved because the FFT of a real sequence demonstrates the following symmetries: Assuming that the FFT output X(k) is complex,

$$X(k) = R(k) + j I(k) \tag{7}$$

and that the sequence has length N, R(k) and I(k) should satisfy the following relations:

$$R(k) = R(N-k), k = 1, ..., N/2-1$$
 (8)

$$I(k) = -I(N-k), k = 1, ..., N/2-1$$
 (9)

$$I(0) = I(N/2) = 0. (10)$$

[†]Improvements have been made and are shown in this table. You may obtain the latest code from the BBS, (713) 274-2323.

In other words, the real part of the transform is symmetric around zero frequency, while the imaginary part is antisymmetric. Similar conditions hold if the transform is expressed in terms of magnitude and phase.

The savings are due to the fact that not all points need to be computed. Since the not-computed points do not need to be saved either, there are also storage savings. An efficient algorithm for real-valued FFTs is described in [10]. This algorithm was implemented in the present study in such a way that, given the sequence of N real numbers $x(0), x(1), \ldots, x(N-1)$, the resulting FFT, consisting of complex numbers, is stored as $R(0), R(1), \ldots, R(N/2), I(N/2-1), I(N/2-2), \ldots, I(1). R(k)$ and I(k) represent the real and imaginary parts of the complex number X(k). Figure 11 shows the memory arrangement for the FFT. Note that the input to the real FFT should be bit-reversed, but the bit reversal can be done as the data is brought in. With this arrangement, an N-point FFT uses exactly N memory locations. If the full array X(k) is needed, the following relations should be used:

$$X(0) = R(0) \tag{11}$$

$$X(k) = R(k) + j I(k), K = 1, ..., N/2-1$$
 (12)

$$X(N/2) = R(N/2) \tag{13}$$

$$X(k) = R(N-k) - j I(N-k), k = N/2+1, ..., N-1$$
 (14)

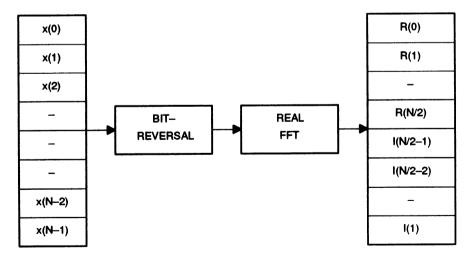


Figure 11. Memory Arrangement of a Real FFT.

It is expected that, in most signal processing applications, there will be no need to reconstruct the full X(k) array and that the output shown in Figure 11 will be sufficient for any further processing.

Appendix C contains TMS320C30 routines implementing a radix-2 real FFT and its inverse. The implementation of the forward transformation is based on the FORTRAN programs contained in [10]. The inverse transformation assumes that the input data are given in the order presented at the output of the forward transformation and produces a time signal in the proper order (i.e., bit-reversing takes place at the end of the program). Viewed another way, the inverse real FFT operates as shown in Figure 11 but with the arrows reversed (and inverse FFT taking the place of the FFT).

The timing for the real-valued FFT (both forward and inverse) is included in Table 2, and the corresponding program sizes are shown in Table 1. As you can see, the real-valued FFT is considerably faster than the corresponding complex FFT because not all the computations need be performed. Furthermore, there are data storage savings because only half the values must be stored. As a result, the maximum length of real-valued FFT that can be implemented on the TMS320C30 without using any external memory is 2048 points. Of course, if all the values are needed, they can be recovered using the symmetry conditions mentioned earlier. To achieve the efficiencies of real FFT and not use any extra memory locations during the computation, the decimation-in-time method is applied [10]. Decimation in time requires the bit-reversal operation in the forward transform to be performed at the beginning of the program rather than at the end. The reverse is true for bit-reversing in the inverse transform.

The Discrete Hartley Transform

Another transform that has attracted attention recently is the Discrete Hartley Transform (DHT)[11, 12]. The DHT is applicable to real-valued signals and is closely related to the real-valued FFT. Comparison of references [10] and [12] describing the implementation of the two algorithms on FORTRAN programs shows that their implementation on the TMS320C30 should be similar. And indeed, this is the case.

The DHT pair is defined for a real-valued sequence x(n), $n = 0, \dots, N-1$, by the following equations:

$$H(k) = \sum_{n=0}^{N-1} x(n) \cos(2\pi k \ n \ / \ N), \ k=0, \ldots, N-1$$
 (15)

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} H(k) \cos(2\pi k \ n \ / \ N), \ k=0, \dots, N-1$$
 (16)

where cas(x) = cos(x) + sin(x). The DHT demonstrates a symmetry that is convenient for implementations: The same program can be used for both the forward and the inverse transforms, and the result is correct within a scale factor. Also, the real FFT and the DHT can be derived from each other [12].

A radix-2 Hartley transform was implemented on the TMS320C30, and the corresponding code is included in Appendix D. This code follows the structure of the real FFT in Appendix C. Tables 1 and 2 show the program memory requirements and the timing for the execution of Hartley transforms of different sizes. The sine/cosine table sizes are the same as in the case of a real FFT.

The Discrete Cosine Transform

The Discrete Cosine Transform (DCT), since its introduction in 1974 [13], has gained popularity in speech and image processing applications because of its near-optimal behavior. This discussion is based on the paper by Lee [14]. The DCT code was developed and implemented by Paul Wilhelm of the University of Washington.

If x(n), $n=0, \ldots, N-1$ is a time-domain signal and X(k) is the corresponding DCT, x(n) and X(k) are related by the following equations:

$$x(k) = \frac{2}{N} \sum_{n=0}^{N-1} e(k) x(n) \cos \frac{(2k+1)\pi n}{2N}$$
 (17)

$$x(n) = \sum_{k=0}^{N-1} e(k) \ X(k) \cos \frac{(2k+1)\pi n}{2N}$$
 (18)

$$e(0) = 1/\sqrt{2} (19)$$

$$e(k) = 1, \quad \text{for } k \neq 0 \tag{20}$$

Appendix E shows an implementation of the DCT based on the paper by Lee [14]. The appendix contains the algorithms for both the forward and the inverse transformations and an example of a table for a 16-point DCT. Note that, because of the structure of the algorithm, the cosine table needed contains actually the inverses of the cosines (within a scale factor), and it is not stored in the natural order. Instead, it is generated by the following C pseudocode:

```
for \{k=2, i=0; k=N/2; k^*=2\}

for \{j=k/2; j< N/2; j+=k\}{

cos\_table[i++] = 1/[2*cos[j*pi/(2*N]]];

cos\_table[i++] = 1/[2*cos[[N-j]*pi/[2*N]]];

\}

cos\_table[N-2] = cos[pi/4];

cos\_table[N-1] = 2/N;
```

The last entry to the table is not part of the cosine itself; it is a constant that is used by the algorithm, and it is placed at the end of the cosine table for convenience.

Table 3 shows the timing of the forward and inverse transforms for different transform lengths. The difference in the timing between the forward and the inverse transforms is due to the fact that more time was expended to optimize the performance of the inverse transform. Since four of the smallest butterflies were done simultaneously in the center program loop, the minimum permissible array size to be transformed is 8.

Transform Size	Forward Transform	Inverse Transform
16	0.019	0.017
64	0.875	0.073
128	0.192	0.161
256	0.418	0.347
512	0.912	0.754
1024	1.982	1.652

Table 3. DCT Timing in Milliseconds

Other Related Transforms

In addition to the FFT types mentioned earlier (complex, real, decimation-in-time, decimation-in-frequency, etc.), newer forms of the FFT have been developed to reduce the computational load. One of the latest in the literature is the *Split-Radix* FFT. The Split-Radix FFT [16] has the lowest number of multiplies and adds of any known algorithm. It achieves this efficiency by combining certain radix-2 and radix-4 butterflies, but, as a result, the classical concept of FFT stages is lost. The new structure uses a rather complicated indexing scheme, which is the price paid for the reduced multiplies/adds. Since, on the TMS320C30, multiplies/adds are not more expensive computationally than any other operation, the indexing scheme wipes out the gains of the reduced arithmetic. Actually, an implementation of the split-radix FFT showed it to be slower than the radix-2 FFT, one of the main reasons being that the block-repeat structure could no longer be used effectively.

Very often, there is a question on what the different benchmark numbers mean. A useful comparison of execution times for different algorithms on different machines has been made [17]. Table 4 presents a small segment of the resulting information that is relevant to the present discussion: the timing in seconds for the radix-8, mix-radix, and split-radix algorithms that were implemented on various machines. Different operating systems and compilers have been used, as shown. The execution times of Table 4 should be compared with the 0.0010055 s that it takes to implement a 1024-point, radix-2, real FFT on a TMS320C30. As can be seen, the TMS320C30 compares favorably to all the other machines investigated.

Table 4. Execution Times in Seconds for a 1024-Point Real FFT. The Numbers Should Be Compared with 0.001055 s of a 1024-Point Real FFT on the TMS320C30

Machine	Radix-8	Mix-radix	Split-radix
VAX 750 UNIX BSD4.2 f77	0.3634	0.3902	0.3021
VAX 750 UNIX BSD4.2 f77 -0	0.2376	0.2948	0.2089
VAX 750 UNIX BSD4.3 f77	0.2545	0.2600	0.2371
VAX 750 UNIX BSD4.3 f77 -0	0.1825	0.2127	0.1672
VAX 785 ULTRIX f77	0.1046	0.1107	0.1101
VAX 785 ULTRIX f77 -0	0.0796	0.0943	0.0811
VAX 785 VMS FOR/NOOPTM	0.0767	0.0871	0.0975
VAX 785 VMS FOR/OPTM	0.0539	0.0641	0.0633
VAX 8600 VMS FOR/OPTM	0.0217	0.0243	0.0235
MICROVAX VMS FOR/NOOPTM	0.1671	0.1846	0.1864
MICROVAX VMS FOR/OPTM	0.1299	0.1527	0.1419
DEC-10 TOPS-10 FOR/NOOPTM	0.0940	0.1184	0.0991
DEC-10 TOPS-10 FOR/OPTM	0.0885	0.1110	0.0845
CDC 855 FTN5,OPT = 0	0.0277	0.0319	0.0338
CDC 855 FTN5,OPT = 1	0.0277	0.0316	0.0337
CDC 855 FTN5,OPT = 2	0.0182	0.0171	0.0151
CDC 855 FTN5,OPT = 3	0.0180	0.0173	0.0150
SUN 3/50 UNIX BSD4.2 f77 - O -f68881	0.2518	0.3365	0.2103
SUN 3/50 UNIX BSD4.2 f77 - f68881	0.2806	0.3897	0.2802
SUN 3/50 UNIX BSD4.2 f77 - O	0.7586	1.047	0.6955
SUN 3/50 UNIX BSD4.2 f77	0.7476	1.029	0.7033
SUN 3/160 UNIX BSD4.2 f77	0.6037	0.6895	0.5660
SUN 3/160 UNIX BSD4.2 f77 - pfa	0.0983	0.1060	0.0946
SUN 3/260 UNIX BSD4.3 f77	0.3689	0.4126	0.3390
SUN 3/260 UNIX BSD4.3 f77 - O	0.3530	0.4142	0.3297
Pyramid 90X UNIX BSD4.2 f77 -0	0.2053	0.2244	0.1416
Pyramid 90X UNIX BSD4.2 f77	0.2206	0.2457	0.1326
HP-1000 21MX-E FTN7X	0.9400	1.248	0.9478
Apple MAC Microsoft FOR	2.6670	3.1600	2.8260
AST PC Microsoft FOR	1.5040	2.0800	1.4630

The TMS320C30 C Compiler

The C compiler for the TMS320C30 permits easy porting of high-level language programs to the DSP device. If the CPU loading of a particular application is not very high, the C compiler can create programs that run on the TMS320C30 in real time. If, however, the result is non-realtime, it may be necessary to use assembly language for more efficient coding.

In most cases, only a portion of the code needs to be written in assembly language. Typically, there are a few code segments where the device spends most of the time and which, when optimized in assembly language, yield the necessary performance improvement. By following the conventions outlined in the run-time environment of the C compiler [15], you can write these time-critical routines in assembly language and call them in a C program. This is also true for the FFT routines. In appendices A, B, and C, the radix-2, radix-4, and real FFT routines mentioned earlier are also put in a C-callable form by adding the necessary interface at the beginning and the end of the code. The tables with the sines and cosines are again assumed to be supplied during link time.

Issues in FFT Implementation

There are many ways of actually implementing the FFT code (and the other transformations), taking into consideration the different possibilities of program locations, the data locations, the ways of input and output, etc. Since it is impractical to cover every possible case, this report has concentrated on a configuration in which the use of external memory is minimized. With the source code and additional explanations provided, you should be able to customize the FFT implementation for a particular application.

Use of External Memory

In these implementations, only on-chip memory was used, and that's why the maximum transform size considered was 1024 points long (2048 for a real transform). Often, though, applications call for use of external memory for program or data or both. When external memory is used, the structure of the code does not change at all; it is only the timing that may be affected.

Fast external memory can be selected so that no wait states are necessary. But even when there are no wait states, accessing external memory may impose some limitations. For instance, you can make only one external memory access in a full cycle, but you can make two accesses of internal memory in each cycle. Also, because of mutliplexing of the busses, pipeline conflicts may arise if both program and data are placed on the same external port. Resolution of such conflicts causes extra cycles for the execution. The section on pipelining in the *TMS320C30 User's Guide* explains in detail what kind of potential conflicts may occur.

To minimize or avoid such conflicts, there are some simple steps that the designer can take. The TMS320C30 has three separate memory areas (one on-chip, one accessed by the primary bus, and one accessed by the expansion bus) that can be combined. For instance, the program can be placed on the expansion port and the data on the primary port. Or the data can first be brought into internal memory and then operated upon. Alternatively, the program may be relocated to internal memory. A related approach is to use the cache. All the transforms are implemented as loops that are executed many times. If you activate the on-chip cache after the first access of the code, the instructions execute from the cache instead of the external memory.

If there are additional conflicts, they can typically be resolved by some rearrangement of the code. For instance, consecutively writing to external memory takes two cycles per write. If, however, a write is followed by some internal operation, then the second cycle of the write is transparent, and the actual cost is one cycle.

Bit Reversal

The TMS320C30 has a special form of the indirect addressing mode for the bitreversing operation that is required at the beginning or the end of an FFT. Through this addressing mode, the scrambled data are accessed in their proper order. This addressing mode works as follows:

Let ARn (n=0..7) be the auxiliary register pointing to the array with scrambled data. The index register IR0 contains a number equal to one-half the size of the FFT. Then, after every access of the data, ARn is incremented by IR0 using the construct

This causes the contents of ARn to be incremented by the contents of IR0, but if there is a carry in this incrementing, the carry propagates to the right instead of to the left. The result is the generation of the addresses in a bit-reversed order. The bit-reversed addressing mode works correctly if the array with the data is aligned in memory so that the first memory address is a multiple of the FFT size. This can be achieved if the first memory address has zeros for the last M bits, where $M = log_2N$, with N being the FFT size. For example, in the case of a 1024-point FFT, the last 10 bits of the memory address of the first datum should be zeros.

In the implementation of the complex FFT, the output is complex even when the input is real. So, there is a need to consider both the real and the imaginary parts of the data array. The above description of the bit-reversed addressing mode assumed that the real and the imaginary parts are stored as separate arrays in the memory. In this case, each of the arrays (real or imaginary parts) can be accessed as described. However, in most cases (including this report), the real and imaginary points alternate in the same array.

In this arrangement, the following simple modification achieves the same goal: set IR0 equal to N instead of N/2, and access the N points of the transform. At every access, the auxiliary register is pointing to the real part of the FFT. The imaginary part is located in the next higher location, and it can be easily accessed.

With the bit-reversed addressing mode, the unscrambling of the data can take place when the FFT result is accessed for further processing or for I/O. It is possible, though, that certain applications demand the reordering of the data in the same array. Such a rearrangement can be done very simply for a complex FFT with the following code.

; DO THE BIT-REVERSING EXPLICITLY

```
LDI
            @FFTSIZ,RC
                           : RC = FFT SIZE
      SUBI
            1,RC
                           ; RC SHOULD BE ONE LESS THAN DESIRED #
      LDI
            @FFTSIZ.IRO
                           : IRO = FFT SIZE
      LDI
            @INPUT,ARO
      LDI
            @INPUT.AR1
      RPTB BITRV
      CMPI AR1.ARO
                            EXCHANGE LOCATIONS ONLY
      BGE
            CONT
                               IF AROAR1
      LDF
            *ARO,RO
11
      LDF
            *AR1.R1
                               EXCHANGE REAL PARTS
      STF
           RO,*AR1
\parallel
      STF
           R1,*AR0
      LDF
            * + ARO.RO
11
      LDF
            * + AR1.R1
                               EXCHANGE IMAGINARY PARTS
      STF
           RO,*+AR1
      STF
           R1.*+AR0
Ш
CONT NOP
           *AR0 + +(2)
BITRY NOP
           *AR1 + + (IR0)B
```

Note that AR1 is pointing to the bit-reversed version of the address contained in AR0. For real-valued FFT, or for FFTs that store the real and the imaginary parts in separate arrays, the real-FFT routine in Appendix C contains a modified example of the above code.

Use of DMA

If the signal to be transformed arrives as a continuous stream of data, the DMA could be used to collect the new data while the data already collected are processed. In this case, the data source address of the DMA points to the memory location corresponding to a serial port, or to another port associated with an external device. The destination is a memory space designated for storage.

There are two ways to use such buffers. One possibility is to designate one buffer as the temporary storage and the other buffer as the working area. When the storage buffer receives the necessary amount of data, the data is transferred to the working area, and the DMA starts refilling the storage buffer. Alternatively, the two buffers are considered equivalent: when the processor finishes processing and outputting the data from one and the DMA has filled the other, the two buffers switch functions; i.e., the DMA starts filling the first buffer while the CPU is processing the data in the buffer just filled.

Test Vector

For testing purposes, a vector with 64 (quasi-random) data points and the corresponding FFT values is given in Appendix F. In this way, if any of the routines is implemented, the test vectors can be used to verify the correct functionality of the routines. Together with the test vectors, Appendix C gives a sine/cosine table for a 64-point transform, and the linking file for such a transform.

Summary

This report examined implementations of fast transforms on the Texas Instruments TMS320C3x floating-point devices. The transforms considered were several forms of the FFT, the Discrete Hartley Transform, and the Discrete Cosine Transform. Because of the powerful architecture of the device, the implementation was done easily and efficiently. It was shown that a TMS320C30 executes the FFTs several times faster than large computers such as VAX and SUN workstations. With the availability of the C compiler, these routines can be put in C-callable form and be used to compute the corresponding transforms efficiently.

Appendices

Appendices A to F contain the TMS320C30 assembly language programs for the different algorithms considered. The contents of the appendices are as follows:

Appendix A: Radix-2 Complex FFT. composed of

A1: Generic Program to Do a Looped-Code Radix-2 FFT Computation on the TMS320C30.

A2: fft_2 - Radix-2 Complex FFT to Be Called as a C Function.

A3: Complex, Radix-2 DIT FFT - R2DIT.ASM.

A4: Complex, Radix-2 DIT FFT - R2DITB.ASM.

A5: TWID1KBR.ASM - Table with Twiddle Factors for a FFT up to a Length of 1024 Complex Points.

Appendix B: Radix-4 Complex FFT. composed of

B1: Generic Program to Do a Looped-Code Radix-4 FFT on the TMS320C30.

B2: fft_4 - Radix-4 Complex FFT to Be Called as a C Function.

Appendix C: Radix-2 Real FFT. composed of

C1: Generic Program to Do a Radix-2 Real FFT Computation on the TMS320C30.

C2: fft_rl - Radix-2 Real FFT to Be Called as a C Function.

C3: Generic Program to Do a Radix-2 Real Inverse FFT Computation on the TMS320C30.

Appendix D: Discrete Hartley Transform. composed of

D1: Generic Program to Do a Radix-2 Hartley Transform on the TMS320C30.

Appendix E: Discrete Cosine Transform. composed of

E1: A Fast Cosine Transform.

E2: A Fast Cosine Transform (Inverse Transform).

E3: FCT Cosine Tables File.

E4: Data File.

Appendix F: Test Vectors, 64-Point Sine Table, Link Command File. composed of

F1: Example of a 64-Point Vector to Test the FFT Routines.

F2: File to Be Linked with the Source Code for a 64-Point, Radix-4 FFT.

Raulx-4 FFI.

F3: Link Command File.

The first three appendices contain the code for the radix-2, complex radix-4, and real radix-2 FFT transformations. These routines are given in both the regular form and in a C-callable form. Furthermore, the contents of a file with the twiddle factors are given, as well as an example of a link command file for a 64-point FFT. Note that the source code of these routines can be downloaded from the TI DSP bulletin board (BBS) by calling (713) 274-2323. For questions regarding the BBS, call the TI DSP hotline at (713) 274-2320.

Acknowledgements

Mr. Raimund Meyer and Mr. Karl Schwarz (Lehrstuhl fur Nachrichtentechnik, University of Erlangen) provided the fast routines of Appendix A to do 1024-point, radix-2, DIT FFT. Mr. Paul Wilhelm of the University of Washington provided the routines for the Fast Cosine Transform (FCT) together with the related explanations and the test vector in Appendix E. Their contributions are gratefully acknowledged.

References

- [1] Burrus, C. S., and Parks, T.W. DFT/FFT and Convolution Algorithms, John Wiley and Sons, New York, 1985.
- [2] Lin, K. -S., Ed. Digital Signal Processing Applications with the TMS320 Family, Prentice-Hall, Englewood Cliffs, New Jersey, 1987.
- [3] Oppenheim, A. V. and Schafer R.W. Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975.
- [4] Rabiner, L.W., and Gold, B. Theory and Application of Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975.
- [5] Burrus, C.S. "Unscrambling for Fast DSP Algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-36, No. 7, pp. 1086—1087, July 1988.
- [6] Papamichalis, Panos E., and Burrus, C.S. "Conversion of Digit-Reversed to Bit-Reversed Order in FFT Algorithms," *Proceedings of 1989 IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 1989.
- [7] Third-Generation TMS320 User's Guide, Texas Instruments, Inc., Dallas, Texas, August 1988.
- [8] Papamichalis, Panos E., and Simar, Ray Jr. "The TMS320C30 Floating-Point Digital Signal Processor," *IEEE Micro*, Vol. 8, No. 6, pp. 13—29, December1988.
- [9] TMS320C30 Assembly Language Tools User's Guide, Texas Instruments Inc., Dallas, Texas, July 1987.
- [10] Sorensen, H.V., Jones, D.L., Heideman, M.T., and Burrus, C.S. "Real-Valued Fast Fourier Transform Algorithms", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-35, No. 6, pp. 849—863, June 1987.
- [11] Bracewell, R.N. "The Fast Hartley Transform," Proceedings of IEEE, Vol. 72, No. 8, pp. 1010—1018, August 1984.
- [12] Sorensen, H.V., Jones, D.L., Burrus, C.S., and Heideman, M.T. "On Computing the Discrete Hartley Transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-33, No. 4, pp. 1231—1238, October 1985.
- [13] Ahmed, N., Natarajan, T., and Rao, K.R. "Discrete Cosine Transform," *IEEE Transactions on Computers*, Vol. C-23, pp. 90—93, January 1974.
- [14] B. G. Lee, "FCT A Fast Cosine Transform," Proceedings of 1984 IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 28A.3.1—28A.3.4, March 1984.
- [15] TMS320C30 C Compiler Reference Guide, Texas Instruments Inc., Dallas, Texas, December 1988.

- [16] Sorensen, H.V., Heideman, M.T., and Burrus, C.S. "On Computing the Split-Radix FFT," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 1, pp. 152—156, February 1986.
- [17] Sorensen, H.V. and Burrus, C.S. "Computer Dependency of FFT Algorithms", *Proceedings of ASILOMAR*, 1987.
- [18] Schuessler, H.W., Meyer, R., and Schwarz, K. "FFT Implementation on DSP Chips—Theory and Practice," Proposal for the 1990 IEEE International Conference on Acoustics, Speech, and Signal Processing.

Appendix A. Radix-2 Complex FFT

Appendix A1. Generic Program to Do a Looped-Code Radix-2 FFT Computation on the TMS320C30

R7, ARO, AR2 ; AR2 POINTS TO XIL.) AR7, RC ; RC SHOLLD BE ONE LESS THAN DESIRED #		BUX1 e4R0 s4R2 R0 · RC=X(1)+X(1)	#AR2++, #AR0++, R1	##K2, ##R0, R3	R2, +480	R3,+4472— ; Y(L)=K3 R0.+480+(1R0) · X(1)=R0.4401		THIS IS THE LAST STAGE, YOU ARE DONE		ecuari, ma Endi	8		•	PSINTAB, ARC ; INITIALIZE IA INDEX (PRC=IA)			-	RO, HRO, HRZ ; (ARL), FULNIER ARZ-RC	1, RC ; RC SHOULD BE ONE LESS THAN DESIRED #	*AR4, R6 ; R6=SIN	٩		**************************************	RZ. R6. R0	*+AR2, *+AR0, R3	R1, *+4R4(IR1), R3 ;	13,±+480	RU, KS, K4 ; K4-RU = LUC-RUS-RUS RI RES RI R	82	8	R3, #PRO++(IRO) ; X(I)=X(I)+X(L) AMD ARO-4RO+2*NI BA BA BA BA . DK-DA-CG-401*CIN	•- •	R4, 4+4472 ; Y(L)=R1+COS-R2+SIN	R7, PR1	
ADDI LDI SUBI	* FIST L00P	# 8P0A	18 S	ABTS		BLK1 STF		* IF THIS IS	-	028	* MAIN TARED - DOG		<u> </u>			ADDI	ADDI		SUBI	*	duu muuso +	RPTB		J.A.	:: ADDF		±8 8	SAGE.	- FDD-		STF	BLK2 STF		I dio	K
GENERIC PROGRAM TO DO A LODPED-CODE PADIX-2 FFT COMPUTATION ON THE		THE PROGRAM IS TAKEN FROM THE BURBLE & PARKS BOOK, P. 111. THE (COMPLEX) MAIN RESIDE IN MINERAL BROWN, THE COMPUTATION IS DONE IN-PLACE BUTTHE BOOK OF THE WARKS TO AMAZINE BURBLE STORTER. TO REPORT THE WARKS TO AMAZINE BURBLE STORTER.	ADDRESSING, THE TAIDLE FACTORS ARE SUPPLIED IN A TABLE PUT IN A LIGHT	SECTION. THIS DATA IS INCLUDED IN A SEPARATE FILE TO PRESERVE THE CENERIC MATIRE OF THE PRINCIAL FOR THE CAME PERPOSE. THE CITE OF THE CET IN AM	LOSZIN) ARE DEFINED IN A JOLOBL DIRECTIVE AND SPECIFIED DURING LINKING.		JULY 16, 1987	· FATRY POINT FOR EXECUTION	FFT SIZE	; LOGZ(N) ; ADDRESS OF SINE TABLE	ATAM TIMES LITTLE VOCASTA	: MEMORY WITH OUTPUT DATA					; STARTING LOCATION OF THE PROGRAM	THE SOUTH OF BUILDING STATE OF THE STATE OF					COMMAND TO LAST BATA DACK POLICE	CONTRACT TO COMO DATA PAGE POINTER		; IRI=N/4, POINTER FOR SIN/COS TABLE	: ANG HOLLIS THE CURRENT STAGE NUMBER	: IRC=2+N1 (BECAUSE OF BEAL/IMAG)	: R7±12	; INITIALIZE REPEAT COUNTER OF FIRST	; LULP . INITIALIZE IS INDEX (ADS=15)	STATE STATE AND ASSESSED ASSESSED ASSESSED.		; CURRENT FFT STAGE	: AND PULMES ID ALL!
YOGRAM TO DO A LOOPED-	_	IAH IS TAKEN FROM THE B IDE IN INTERNAL MENORY.	G. THE THIDDLE FACTORS	THIS DATA IS INCLUDED THE PRINCHAM FOR THE	RE DEFINED IN A . GLOBL	ATTHREE PANCE E PARAMETER ST.	TEXAS INSTRUMENTS	FF.T		Be and a sinker	100 mil.				w		D FFT	001		z 1	SING.	D 00.TP	CCTC17	700	GFFTS17, 1R1	-2, IRI	0, PK6	1. IRO	@FF1S17,R7	1,487	98		•	#++AR6(1)	CINE UI, MENU
* GENERIC F	+ TMS320C30.	THE PROGRAM	ADDRESSI	MATING OF	1062(N) 4	0 - AUTHOR: 0	-	8 03	.GLOBA	10.09 10.09 10.09	1.7331		-	TEXT.	INITIALIZE		. HORD	30405		HISIZ 1000		DUTPUT . MORD	201.		ig	<u>ਡ</u>	<u> </u>	3 3	ē	i i	187	i	OUTER LOOP	.00F: MOP	3

; RC*N ; RC SHOULD BE ONE LESS THAN DESIRED \$; IRO=SIZE OF FFT™ 1 INCREMENT LOOP COUNTER FOR NEXT THE ; IE=2+IE ; N1=N2 ; N2=N2/2 ; NEXT FFT STAGE + STORE RESULT OUT USING BIT-REVERSED ADDRESSING
+ END: LDI BETISIZ.RC · DY-L *+ARO(1), RO *ARO++(IRO)B, R1 RO, *+AR1(1) R1, *AR1++(IR1) OFFISIZ, RC 1, RC OFFISIZ, IRO 2, IR1 EINPUT, AR0 EQUIPUT, AR1 ëëëë**ë**ë ₹ <u>5</u> 5 5 :: BI :: * SE.F.

; BRANCH TO ITSELF AT THE END

SE.F

Appendix A2. fft_2-Radix-2 Complex FFT to Be Called as a C Function

. 6.008	1. 1. 1. 1. 1. 1. 1. 1.					2	
- GLOBEL - 1510** - LEAS FETSIZ. 1 - LESS LOOPFT. 1 - LOOPFT.	- GLOBE - 1510** - LESS FF1512.1 - LESS 100°FT.1 - LESS 100°FT.1 - LESS 100°FT.1 - LESS 100°FT.1 - LEST 510°FT.1 - LEST 510°FT.1 - LEST 60°FF 60			•			
SES STRIZE	SESS FF1511.1	PSIS: NT 6ft 2(N. M. DATA)			8 000	-##.2	: ENTRY POINT FOR EXECUTION . ANDRESS OF STAFF TABLE
*** FFTS12.1 *** SS 100FFT.1 *** SS 100FFT.1 *** SS 100FFT.1 *** INITIALIZE C FUNCTION *** INITIALIZE C FUNCTION *** CASH *** PASH ** PASH *** PASH	*** FFTS12, 1 *** SS			•			THE STATE OF STATE
** ** ** ** ** ** ** ** ** ** ** ** **	** ** ** ** ** ** ** ** ** ** ** ** **				SS	FFTS17.1	
** TEXT ** INITIALIZE C FUNCTION ** INITIALIZE FFT BOUTHE ** INITIA	** TEXT ** SINTAB word ±5106 ** INITIALIZE C FUNCTION ** INITIALIZE C FUNCTION ** INITIALIZE C FUNCTION ** INITIALIZE C FUNCTION ** INITIALIZE F RA ** PROST RA		UT DATA		8	L00FFT, 1	
TEXT SINTAB	SINTAB	RIPTION		•	2	1,10	
SINTAB . wordsine - INITIALIZE C FUNCTION	SINTAB	ENERIC FUNCTION TO DO A RADIX-2 FFT CO	PUTATION ON THE 320C30.		.TEXT		
SINTING	SINTING	HE DATA APPRAY IS 244-LONG, WITH REAL A	ND INSGINARY VALUES ALTERNATING.	•			
MITIALIZE C FONCTION	## INITIALIZE C FUNCTION ## CASH PASSH PA	he program is based on the fortham pro	HOW IN THE BURRUS AND PARKS	SINTAB	8016	_sine	
## 1984 FP ## 198	## 1.5 PUSM FP			INI +	IALIZE C	FUNCTION	
## 25 P NSS		HE COMPUTATION IS DONE IN PLACE, AND 1	E ORIGINAL DATA IS DESTROYED.	•			
PUSSH RS PASSH PASSH RS PASSH	Pussi	IT REVERSAL IS IMPLEMENTED AT THE END ECFSSARY THIS PART CAN BE COMPENTED O	DE THE FUNCTION. IF THIS IS NOT	-Ht-2:	<u> </u>	£ 9	; SAVE DEDICATED REGISTERS
P. CSSH RS P. CSSH P.	P. C.S.H				3	: : 2	
PASSF R6 PASSF R6 PASSF R6 PASSF R65	PASSF R6 PASSF R6 PASSF R68 PASSF R68 PASSF R68 PASSF R68 PASSF R68 PASSF R67 PASSF R6	HE SINE/COSINE TABLE FOR THE THIDDLE F	ICTORS IS EXPECTED TO BE SUPPLIED		P.CS.	1 2	
PUSH #RV STI NO, EFFTSIT LDI +FPTD, NO STI NO, EMBOTT I MITTALIZE FFT ROUTINE LDI GFFTSIZ, IRI LDI GFFTSIZ, RP LDI LAPS LOOTER LOOP LOOP: NOP FFTSIZ, RP LOOTER LOOP STI NO, MRZ LOOP: NOP FFTSIZ, RP LOOTER LOOP 1, APRS 10 1, AP	PLOSH	URING LINK TIME, AND IT SHOULD HAVE TH	FOLLOWING FORMAT:		PUSHF	28 5	
PUSSH ARRS PUSSH ARRS PUSSH ARRS PUSSH ARRS PUSSH ARRS PUSSH ARRS LDI +FP(2), RN STI RO, EFFSIZ LDI +FP(3), RN STI RO, EINEUT LDI +FP(3), RN STI RO, EINEUT LDI GFFSIZ, IRI LDI 1, ARRS LDOP: NOP H-+ARRS (1) LDOP: NOP H-+ARRS (1) LOOP: NOP H-+	PUSH				1 2	× 5	
PUSSI ANG PUSSI	PUSSH ANG PUSSH				1 2 2 3	ž 4	
PUSH AR7 101 +FP(2), R0 511 R0, defF1512 101 +FP(3), R0 511 R0, defF1512 101 +FP(4), R0 511 R0, defF1512 101 +FP(4), R0 101 HFP(4), R0	LDI +FP(2), RO		(N/10#)		Æ	98	
101 +FP(2),R0 511 R0, 6FF512 101 +FP(4),R0 511 R0, 6L05F7 101 +FP(4),R0 511 R0, 6L05F7 101 +FP(4),R0 511 R0, 6L05F7 101 +FP(4),R0 101 GFF512,R1 101 GFF512,R1 101 GFF512,R1 101 GFF512,R1 101 GFF512,R1 101 1,APS 101 1,APS 101 R1,APS 101 R1,APS 101 R1,APS 101 R1,APS 102 R1,APS 103 R1,APS 104 R1,R0 105 R1,APS 106 R1,APS 107 R1,R0 107 R1,APS 108 R1,APS 109 R1,APS 10	### CDI + FF(2),80 \$11				PUSH	784	
111 60, 6F151, NO. 6F151, NO. 511 60, NO. 511	STI RO, #FTST, NO STI RO, #FTST, NO STI RO, #FTST, NO STI RO, #TRO, RTST, RO, RT			•	ā	6 6 6 6 6 7	CHARLES CHARLES OF THE CONTRACTOR STORY
11 17 17 17 17 17 17 17	11 W-PF 131.6 15		(N/Ida7a/I-4		i 5	Do despery	THE MANES IN THE DOCCOME.
STI NO, BLOGFT STI NO, BLOGFT STI NO, EINPUT LUI +FP(4), RO LUI 6FF1SIZ, IRI LUI 0, RAB LUI 0, RAB LUI 0, RAB LUI 1, RAS LUI 1, R	STI NO. BLOGGT. STI NO. BLOGGT. STI NO. BLINUT I LIMITAL.IZE FT ROUTINE LUI GFFSIZ., IRI LUI GFFSIZ., IRI LUI GFFSIZ., IRI LUI GFFSIZ., IRI LUI GFFSIZ., RO LUI L., ARC SUBLI L., RO	TO SEE CALLED CALLED FILE OF THE PARTY.	CAME LIAME UM INC. FOR AM		; =	MV, 47 1312	THE MAKES IN THE TRUMBER
101 +FP(4), RO 511 RO, EINPUT 101 GFT512, IR1 101 GFT512, IR1 101 GFT512, IR1 101 GFT512, IR0 101 GFT512, IR0 101 GFT512, IR0 101 GFT512, RO 101 GFT5	101 +FP(4), R0 111 R0, Elbeuri 111 R0,	POINT FFT. THERE ARE NHW/4 VALUES FOR	A FULL AND A GUARTER PERIOD OF		i IS	RO. BLOGFFT	
INITIALIZE FTT ROUTINE INITIALIZE FTT ROUTINE LUI	INITIALIZE FFT ROUTINE	E SINE MAVE, IN THIS MAY, A FULL SINE	AND COSINE PERIOD ANE AVAILABLE		ë	#-FP(4), RO	
INITIALIZE FFT ROUTINE USA	INITIALIZE FFT ROUTINE	UPERIMPOSED).			STI	RO, EINPUT	
- LDI 66FFS12,1R1 - LSM 2, IR1 - LDI 0,4866 - LDI 0,4866 - LDI 0,4867 - LDI 1,4867 - DI 1,4865 - DUTER LOOP - NOP - ++4486(1) - LOOP: NOP - ++4486	* LDI 6FF1817,181 : 1.84 - 2, 181 LDI 0,886 LDI 0,1867 LDI 0,1867 LDI 0,1867 LDI 0,1867 LDI 0,1867 LDI 0,1866 LDI 0,	ACK STRUCTURE UPON THE CALL:		INI +	IAL IZE FF	T ROUTINE	
UDI 6FTFSIT, IRI LSM - 2, IRI LDI 0, 486 LDI 6FTFSIT, 180 LSM LDI 1, 487 LDI 1, 487 LDI 1, 485	UDI 6FTS15,181 USH - 2, 181 UDI 0,496 UDI 6FTS12,180 UDI 6FTS12,180 UDI 1,487 • OUTER LOOP UDI 61P401,480	-		*			
1 184 - 2, 184 - 1	LUI 0,486 LUI 6FF1SIZ,180 LUI 6FF1SIZ,180 LUI 6FF1SIZ,187 LUI 1,487 LUI 1,48				ë	GFFTS17, IR1	
LDI 0,486. LDI 0,486. LDI 1,487	LDI 0,486. LDI 0,486. LDI 0,486. LDI 1,180 LSH 1,180 LSH 1,180 LDI 1,487 • 0.07ER LOSP • 0.07ER LOSP • 0.07ER LOSP LDI E!HPUT,480 : LDI 62140T,480 : LDI 6214				ऊ	-2, IRI	; IRI=N/4, POINTER FOR SIN/COS TABLE
LDI 6FFFSIZ,180 LSM 1,180 LSM 1,180 LDI 1,485	LUI 6FTS12,180 LS4 LT 1,180 LD1 6FTS12,180 LD1 1,487 LD1 1,485 LD				ë	0, AR6	; AR6 HOLDS THE CURRENT STAGE NUMBER
1 1,180 1 1,180 1 1,485 1 1,180 1 1,485 1 1 1 1,485 1 1 1,485 1 1 1,485 1 1 1,485 1 1 1,485 1 1 1,485 1 1 1,485 1 1	LSH 1,180 LDI (#FTSI7,87) LDI 1,487 LDI 1,487 LDI 1,487 LDI 1,487 LDI 1,487 LDI 618401,80 LDI 618401,80 LDI 618401,80 LDI 67,80,982 LDI 897,80,982 LDI 897,80				5	BFFTSIZ, IRO	
LUI 0477 FISIL, R7 LUI 1, AR5 LUI	LUI 0477 7 1 1,487 1 1 1,487 1 1 1,487 1 1 1,487 1 1 1 1,487 1 1 1 1,487 1 1 1 1,487 1 1 1 1,487 1 1 1 1 1,487 1 1 1 1 1,487 1 1 1 1 1 1,487 1 1 1 1 1 1 1,487 1 1 1 1 1,87 1 1 1 1,87 1 1 1 1 1,87 1 1 1 1,87 1 1 1 1,87 1 1 1 1,87 1 1 1 1,87 1 1 1 1,87 1 1 1 1 1,87 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				3 :	1,180	; IRO=24N1 (BECAUSE OF REAL/INAG)
101 1,485 101 101 1,485 180 180 180 180 180 180 180 180 180 180		† 			5	OFFTS12,R7	; R7=R2
200 1 10 1 1,485 1 10 1 1 1,485 1 10 1 10 1 10 1 10 1 10 1 10 1 10 1	LUI 1,445 • OUTR LOD* • OUTR	SECTION INSTITUTE OF THE PROPERTY OF THE PROPE	ă	•	Ē.	, I	; INTITIALIZE REFERI COUNTER OF FIRST
- 07/15R LOOP - 100/20 - 07/15R LOOP - 100/20 - 07/15R LOOP - 100/20 - 07/15R LOOP - 100/20 - 100/20 -	* 0.07ER L.00P * L.00P: N.0P +++496.(1) ; L.01 [HEVI]480 ; L.01 [487, RC] ; S.161 1, RC] ;	AR6, AR7, IR0, IR1, RS,	نو يو	•	Ē	1,485	: INITIALIZE IE INDEX (ARS=IE)
1009: 1009: 1009: 1009: 1009: 1009: 1009: 1001 1001	1009: 1.0	THOR: PANCS E. PAPANICHALIS		• •	007		
100 - 100 -	100*: NUT ***********************************	_	ER 13, 1987	*	ş		
101 A7	10.1 (1970) 1970 1970 1970 1970 1970 1970 1970 1970			:007	2 :	***************************************	; CURRENT FFT STACE
AR7, RC	AR7, RC 1, RC	***************************************	*******************		9 6 6	R7.480.482	: ARC POINTS TO X(L)
	 E.,				198	AR7, RC	A CHARLOS MALE OFFI PAR OF CAROLO OF

101 R7.1R0 1.88 1.89											
--	--	--	--	--	--	--	--	--	--	--	--

Appendix A3. Complex, Radix-2 DIT FFT-R2DIT.ASM

COMPLEX, RADIX-2 DIT FFT : R2DIT.ASM +	***************************************
+ + 	+ THIS PROGRAM INCUDES FOLLOWING FILES:
ON THE THIS 220C30 +	+ THE FILE 'THIDIKBR. ASM' CONSISTS OF THIDDLE FACTORS
HRITTEN BY: RAININD REYER, KARL SCHAMR? LEMESTUAL FIREN MCCHATCHTENHIK UNIVERSITÄRET ENLANCEH-MEDMERGE CALENSTRASSE 7, D-6520 BLLANZH, FRG *	∃ _ ″
THE (COPPLEX) DATA RESIDE IN INTERNAL REDION, THE COPPUTATION IS DONE # IN-PLACE, BUT THE RESULT IS NOVED TO ANDTHEN REDION SECTION TO # IEDONOSTRATE THE BIT-REVERSED ADDRESSING.	+ ADDRESS (CORFFICIENT) = CORFFICIENT + 0 R(AM(0)) = COR(2PF146/32) = 1 + 1 - [LAM(0)] = SIN(2PF146/32) = 0 + 2 R(AM(4)) = SOR(2PF146/32) = 0,707 + 2 R(AM(4)) = COR(2PF146/32) = 0,707 + 3 R(AM(4)) = COR(2PF146/32) = 0,707 + 3 R(AM(4)) = COR(2PF146/32) = 0,707 + 4 R(AM(4))
FOR THIS PROCESSA THE MINIMAN FFTLENGTH IS 32 POINTS BECAUSE OF THE ** SEPARATE STAGES. **	
FIRST TWO PASSES WEE REALIZED AS A FOUR BUTTEMPLY LODP SINCE THE ** **********************************	• 12 R(WH(3)) = C0(C)PF143(23) = 0,881 • 13 -1(WH(3)) = SIN(2)PF143(23) = 0,556 • 14 R(WH(7)) = C0S(2)PF17(23) = 0,195 • 15 -1(WH(7)) = SIN(2)PF17(23) = 0,981
* EXAMPLE FOR A 1024-POINT FFT (EXGLIDING BIT PENERSAL):	* WHEN GENERALED FOR A FFT LENGTH OF 1024, THE THRIE IS FOR ALL A MAILLABLE FFT OF LESS OR EQUAL LENGTH.
= 229 u0ets = 512 u0ets	THE MISSING TAUTOLE FACTORS (MAIL), MAIL),
	TO CHANGE THE FFT LENGTH, ONLY THE PARAMETERS IN THE HEADER OF TAILDIXKIR, ASH AND THE INPUT AND COTPUT VECTOR LENGTIS, MED TO BE ALTERED.
*** **********************************	
	* FR + JB 1 (USS - J SIN) BR' + J BI' * TR = BR + COS + BI + SIN * TI = BR + SIN - BI + COS * #6" = RA + IR * TI = BR + COS + BI + COS
	+ 14/1 -

:: STF R2,4483++ LD1 APS,RC	*	•	•	•	•	•	•	•	•	•	***************************************			. 95 = 81 + 51W (A6) = 95)	1 LO - DI + SIN , (M LO)	. (R) = T1 = B0 - R1)	: R0 = BR + COS . (R3 = AI + II)		; (R4 = AI - TI, BI' = R3)		; R3 = TR = R0 + R5	; RO = BR + SIN , R2 = AR - TR	300	; KI = BI + UAS , (AI' = K4)	. 55	£ .				; R2 = T1 = R0 - R1	; K3 = A1 + 11 , BK′ = K5	. R4 = A1 - T1 B1' = B3		; ADDRESS UPDATE	; R1 = B1 + COS , A1' = R4	;	1. PO = 1990 + SIN	; K3 = IK = K1 - K0 , K0 = BK + COS	\$ 150 min	. H = DI + DIN' NZ = HN - IN :	· B5 = 36 + T6 - B6 / = 87	7				
R2, #483++ AR5, RC		-LY-TYPE:		TR = DR + COS + BI + SIN	TI = BR + SIN - BI + COS	Æ	=-	E -	11 +		***************************************		# CV	8 8 B	D. 400.74	20 80	#AR1, R7, R0	R2, +4R0, R3	R2, +AR0++, R4	R3, #AR3++	80, R5, R3	#4R1++,R6,R0	K3, 49K0, K2	DA ADOL	#400+ R3 R5	R2. #4R3++		O NEXT GROUP		R1, R0, R2	K2, #9K0, K3	R2 +480++(1R1) R4	R3, #4R3++(IR1)	##R1++(IR1)	*4R1,R7,R1	R4, +482++(IR1)	#4R1, R6, R0	ON, ++ / ##K/ ++ 1 KG	MO, KI, K3	D2 4000 D2	#4R0++ R3 R5	F2 #483#	. P			
ST ST		 FIRST BUTTERFLY-TYPE: 		£E.=	£5.=:::	FT + 26 = 26	P AI'= AI	FT - 54 = 75	BI'= AI + TI		***************	-	9	JA OH			J. de	:: ADDF	SUBF	:: STF	ADD		*		PELY1 ANDE	SIF		SHITCH OVER TO NEXT GROUP		18	100		:: STF	ģ		SIF	i i			100		±5				
; AI' = R4 = R2 + R0		; B1′ = R2 = R2 - R0	; Cl' = R3 = R6 + R7	; AI' = R4, BI' = R2		; DI' = R7 = R6 - R7	; DI' = R7 , CI' = R3											; POINTER TO THIDDLE FACTOR	; GROUP COUNTER	, UPPER REAL BUTTERFLY INPUT	: UPPER REAL BUTTERELY OUTPUT	COMEN MEAL BUILDING VINER	; LONEX REAL BUILDINGS	· HALF BLITTERS Y COUNT	CER LSB	; HALF STEP FROM UPPER TO LOWER REAL	; PART		; STEP FROM OLD IMMGINARY TO NEW REAL	: VALUE : NAME ON VERD ANDRESS INDIVITE	; R7 = 005	-		arther in Command in the manage . The	AND = UPPER MEME BUTTERS VINER	. AST = LONGER DECK DISTERSELY CATEGOT	· ARS = 104FR REAL BUTTEREY DUTPUT	THE IMMOJIMMAY PART HAS TO FOLLOW	: R6 = SIN	: R1 = BI + SIN	; DUMY ADDF FOR COUNTER UPDATE	••	; R3 = TR = R0 + R1 , R0 = BR + SIN	0 - 00 - 00 - 00 - 10 - 10	1 L = B1 = CO3 , N2 = M4 - IN	; R5 = AR + TR , BR' = R2
RO, RZ, R4		RO, R2, R2	R7, R6, R3	R4, +4R4	72, #85	R7, R6, R7	R7, #AR6	R3,+#R2			THIRD TO LAST OF STAGE 2		101	IRO ARS	500	986	1	ESINTAB, AR7	0, AR4	EINPUT, ARO	ARO, AR2	JRO, ARO, ARG	1 ADL	20 C-	1.465	-1, IR0		-1, IRI	1, IRI	70 11 100	#4R7, R7								#++4R7,R6	#4R1, R6, R1	**************************************	#AR1, R7, R0	#4RI++, #4R7, R0	RO, R1, R3	10', N', NI	**************************************
A00F		SEE.	8	¥.	STF	B	SIF	STF			OD TO LAS!		=	9	į	į		ë	ë	5 :	5 9	3 3	3 3	5 5	3	35		<u> </u>	100	2	Ė				FILL PIPELINE				¥	H PYF	ADOP	H	HPYF.		± 0	ADD

•••	* * *	• •		***************************************			; P5 = B1 + COS , (AR' = R5)	· (R2 = T1 = R0 + R1)	: RO = BR + SIN , (R3 = AI + TI)		; $(R4 = AI - TI, BI' = R3)$: TR = R3 = R5 - R0	: R0 = BR + COS , R2 = AR - TR	10 - 717 - 110 - 10 - 10 - 10 - 10 - 10	1 1 - D1 4 51N , (HI = N4)	; R5 = AR + TR , BR′ = R2			13 + 68 = 11 = 68 :	; R3 = AI + TI		; DO FOLLOWING 3 INSTRUCTIONS	; R4 = AI - TI , BI' = R3	362 - 28	88 = .10 ·	BRANCH HERE		; JUPP OUT AFTER LD(N)=3 STAGE		; UPPER INPUT
	FILL PIPELINE	1. BUTTERFLY: w^0	A005	ADDF	Mans .	2. BUTTERFLY: w^0	4	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	ADDF	SUBF	# ## ##		±s ::	STF			* 3. BUTTERFLY: WYN.4	y y	ans a	ADDF	SUBF	4. BUTTERFLY: WYH/4) ()		# S# E#	5. TO M. BUTTERPLY:	95 8T	## E# ## E# ##
55, 1R0 6F.08HZ, RC	w	\$	+4R0, +4R1, R2 +4R1++ +4R0++ R3	*ARO, *AR1, R0	##R1++, ##R0++, R1	٤	TO LOVE OFF	#481++ #480++ R7	*ARO, *AR1, R4	#4R1++(IRO), #4RO++	R2, #M22#	RO, #M2++	R1, #AR3++	R6, +AR2++	R4, +482++(IR0)	R5, #4R3++(IR0)	: 4/4/4	30 10010	#4R1 #4R0 R4	*AR1++, *AR0, R6	#AR1++, #AR0++, R7	. 24/4		##\$R1, ###\$R0, R3	# 12 T	#4R1++(IR0), #4R0++, R2	R5,+4R2++ R7,+4R3+	IEBELY:	BFZEND	**************************************
; PUINIEK 10 INLUULE FACTOR ; DISTANCE BETNEEN THO GROUPS			. 146' = 152 = 146 + 196 . 180' = 153 = 146 = 186	; AI' = R0 = AI + BI	; BI' = Ri = AI - BI		94 - 70 - 70 - 70 - 70 - 70 - 70 - 70 - 7	E - 2E -	; AI' = R4 = AI + BI	#4R1++(IRO), #4RO++(IRO), R5 ; BI' = R5 = AI - BI	; (AR' = P2) ; (BR) = P2)	(AI = RO)	; (BI' = RI)	; A6′ = R6	: AI′ = R4	: B1′ = 75		2 · · · · · · · · · · · · · · · · · · ·	AI' = R4 = AI - 99	; BI' = R6 = AI + BR	; BR' = R7 = AR - BI			; AR' = R3 = AR + BI	. BO = BR (FOR INNER LODE)	, RY = R2 = A8 - B1	; (BR/ = R7)	(OL - 10)	:	; R7 = COS , ((A1' = R4))

1	(RZ = T1 = RO + R1)					##K1++(1K0), K6, K1	, K1 = B1 + S1N , K3 = AK + 1K
Fig. = Fig. = May Fig.	RO = RP = COS (RS = AI + TI) RODE	چ	;				
FOR = FOR + COS (FOS = ALI + TI)	RO = RR + COS (RS = AI + TI)		2 = TI = R0 +	RI)	:: ADDF	#ARO++, R3, R3	
Fig. = Ri + COS (AR' = Ri) Fig.	RO = RO = RO = RO RO RO RO RO	 8	. 800 + 861 =	(R3 = AI + TI)	•		
R2 = R8 + R8 R2 = R8 - R8 R2 R3 R2 R4 R4024 R5 = R8 + R8 R2 = R8 - R8 R3 R2 R4 R4024 R5 = R8 + R8 R2 = R8 - R8 R3 R2 R4 R4024 R5 = R8 + R8 R2 R2 R3 R2 R3 R4024 R5 = R8 + R8 R2 R2 R3 R4024 R5 = R8 + R8 R2 R3 R3 R3 R3 R4024 R5 = R8 + R8 R3 R3 R3 R3 R3 R4024 R5 = R8 R3 R3 R3 R3 R3 R3 R3	FR = FR + TR FR - FR FR - FR FR FR FR F		;		+ OLEAR PIP	7.INE	
RO = RP = SIN RC = RP - R	RO = RP = SIN RO = RP RO RO = RP RO RO RO = RP RO RO RO RO RO RO RO		. II - IH = +	B1 = K3)	ŧ	2	: :
RO = BR + SIN RO = RR - TR ABDF R1, R0, RO	RN = BN + SIN RN = RN - RN RN RN = BN + SIN RN = RN - RN RN RN = RN RN RN RN RN	2	3 + 08 = 97 =			17, e863+	, BK = KZ , MI = K4
RG = RI + COS (AI' = RN) 15 15 15 14 14 15 15 15	R	2	NIS + SE =	Z = 48 - TR		200	. B) = T1 = B0 + B1
Fig = Bi + CDS (AI/ = Bi) SIF R2, +460, PM			•		A00.	R2, #\$90, R3	: R3 = AI + TI . AR' = R3
SIBE R2,440, PM 1	Super 1: K5 = Mt + TR , BK = R2 1: K5 = Mt + TR , BK = R2 1: K6 = T1 = R0 - R1 1: R0 = R1 + S1M , (MK = R5) 1: R0 = R4 + TR , BK = R3 1: R1 = R1 + COS , (MK = R3) 1: R1 = R1 + COS , (MK = R3) 1: R1 = R1 + COS , (MK = R3) 1: R1 = R1 + COS , (MK = R3) 1: R1 = R1 + COS , (MK = R3) 1: R1 = R1 + COS , (MK = R3) 1: R1 = R1 + COS , (MK = R3) 1: R1 = R1 + COS , (MK = R3) 1: R1 = R1 + COS , (MK = R3) 1: R1 = R1 + COS , (MK = R3) 1: R2 = R4 + TR , BK = R3 1: R3 = R4 + TR , BK = R3 1: R4 = R4 + R4 + TR , BK = R4 1: R4 = R4 + R4 + TR , BK = R4 1: R4 = R4 + R4 + TR , BK = R4 1: R4 = R4 + R4 + TR , BK = R4 1: R4 = R4 + R4 + TR , R4 + TR , R4 1: R4 = R4 + R4 + TR , R4 + TR , R4 1: R4 = R4 + R4 + TR , R4 + TR , R4 1: R4 = R4 + R4 + TR , R4 + TR , R4 1: R4 = R4 + R4 + TR , R4 + TR , R4 1: R4 = R4 + R4 + TR , R4 + TR , R4 1: R4 + R4 + TR , R4 + TR , R4 1: R4 + R4 + TR , R4 + TR , R4 1: R4 + R4 + TR , R4 + TR , R4 1: R4 + TR , R4 + TR , R4 1: R4 + TR , R4 + TR , R4 1: R4 + TR , R4 + TR , R4 1: R4 + TR , R4 + TR , R4 1: R4 + TR , R4 + TR , R4 1: R4 + TR , R4 + TR , R4 1: R4 + TR , R4 + TR , R4 1: R4 + TR , R4 + TR , R4 1: R4 + TR , R4 + TR , R4 1: R4 + TR , R4 + TR , R4 1: R4 + TR , R4	₹.	= BI + COS ,	(AI '= R4)		K3, *#R2++	
FG = BI + SIN (AR' = FG)	FG = FR + FR FR FR FR FR FR FR			ď	187S		; R4 = AI - TI , BI' = R3
Fig. = Bi + SiN (AR' = FS) UI CHPUT, AND FIG.	FG = B	£		K' = 12		F3, 44R3	; AI′ = R4
(RZ = TI = RO - RI)	(RZ = T1 = RO - R)	8	4 DI 4 CIN	(<u>\$</u>	+		
101 6114-01, 400 111 6114-01, 400 111 600, 402 111 610, 610, 612 111 610, 610, 612 111 610, 610 111 610, 610 111 610, 610 111 610, 610 111 610, 610 111 610, 610 111 610, 610 111 610, 610 111 610, 610, 610 111 610, 610, 610 111 610, 610, 610 111 610, 610, 610 111 610, 610, 610 111 610, 610, 610 111 610, 610, 610, 610 111 610, 610, 610, 610 111 610, 610, 610, 610 111 610, 610, 610, 610 111 610, 610, 610, 610 111 610, 610, 610, 610 610, 610, 610 610, 610, 610 610, 610, 610, 610 610, 610, 610, 610, 610, 610, 610, 610,	(RZ = T1 = RO - R1)	2	, M10 = 10 =	(Cu -	Maria ista		
RO = BR + COS RO = A1 + T1 LD RAD, ROZ	RO = BR + COS (RO = A1 + T1) UDI		2 = TI = B0 -	£	•	ATMENT AND	1000 1000 ·
RA = AI - TI , BI ' = R3	RA = RI - TI , BI' = R3 UDI	2	= BR + C05	(R3 = AI + TI)	i	OF 100	. Happe orman
RR = RI - TI , BI ' = R2 LDI	RA = RI - TI , BI' = R3)		•		jē	DIMPITTP2 AR	THE STATE OF THE S
R3 = R8 = R0 + R5	R3 = R3 = R0 + R3	<u></u>	4 = AI - TI ,	BI′ = R3)	5	ARI AR3	. LOLER GUTPUT
R3 = R8 + S14 R2 = R8 - R8 LD1 S-180	RA = RR + RR RR - RR LDI				9	eSIMTP2 AR7	POINTER TO THIRDILE FACTORS
R1 = B1 + COS (AI' = RA)	R1 = B1 + COS (AI' = RA)	 55	- TR = R0 +	ın	3	3.180	GROUP OFFSET
R1 = B1 + CDS (AI' = R4)	R1 = B1 + COS	 8	. NIS + 25 =	F2 = 48 - 18	ē	BFG4M2, RC	
R3 = R4 = R4 R4 R4 R4 R4 R4 R4	R3 = R4 + R7 BF < R2 1 BUTER-LY1 W R5 R5 = R4 + R7 BF < R3 1 BUTER-LY1 W R5 = R4 + R3 R5 = R4 + R1 R5 = R4 + R1 R5 = R4 + R1 R5 = R4 + R3 R5 R5 R5 R5 = R4 + R3 R5 R5 R5 R5 R5 R5 R5		300	, ed - /14/	•		
Fig. = Fig. + TR Fig. = Fig. 1. BUTTERFLY: w^O	RS = RR + TR BN = RZ		93	(M) * R4)	+ FILL PIPE	X	
FG = B1 = COS (AR' = R3) SUBF		 &	= AR + TR .	R' = R2	+ 1. BUTTER		
10	MS = M1 + COS (MR' = R3) Super				•		
(R2 = T1 = R0 - R1)	(RZ = T1 = RO - R1) AUDF	£2	= BI + COS	(Mg/ = M3)		#4R0, #4R1, R6	. A5. = 75. = 35. + 35
(RZ = TI = RO - RI) SIBF +48(I++(IRO), +48O++(IRO), +46O++(IRO),	(RZ = T1 = RO - R1) Signe				704	+480 +481 R4	. Al' = R4 = Al + Bl
RO = BR + SIM (R3 = AI + TI)	RO = BR + SIN (R3 = AI + TI)	 B	2 = TI = R0 -	71)	187 0	#AR1++(IRO) #ARO++	(IRO), R5 ; BI' = R5 = AI - BI
RA = AI - TI BI / = R3	(RA = AI - TI , BI = R2)	 8	* NIS + SIN	(R3 = A1 + T1)	•		
FR = FR = FR = FR FR FR FR FR	RA = RA - TI , BI = RA Audre Audre Audre RA - RA Banda Audre A				# 2. BUTTER	LY: w/H/4	
MODE +++++	ADDF UDF SUBF SUBF STF TEN TEN TEN TEN TEN TEN TEN T		, II - IE *	BI * K3)	•		
	11. 11. 11. 11. 11. 11. 11. 11. 11. 11.	8	50 - 5T -			**************************************	; AR' = R3 = AR + BI
1	ST S	2 8	2 4	£ .		#47.K	; R1 = 0 (FOR INNER LOOP)
STF RA-#4624- 1100 11	STE			¥ . ¥ . ½		#4R1++,R0	æ
	# 3. TO M. BUTTER # 4. STF # 5. STF # 5. STF # 6. STF # 7. STF #	-	= BI + SIN	(A1 ' = R4)	and the	A THE PARTY OF THE	•
STE 75, +863+1(80) : 1	1917 1917 1917 1917 1917 1917 1917 1918 1919					70° 140° 11	(FE = 16)
* 3. TO N. BUTTENELY: * 3. TO N. BUTTENELY: * LIF ##87+1R7 ; STF R4.#862+1(R0) ; STF R4.#862+1(R0) ; STF R3.#863+ R8 ; R3	8 3. TO 11. BUTTER 1 1. LDF 1 2. STF 1 2. STF 1 3. TO 11. BUTTER 1 4. TO 11. BUTTER 1 4. TO 11. BUTTER 1 5.	12	= AR + TR .	R' = R2		77, menore	(N) is
# 3. TO N. BUTTERSTLY: 1	* 3. TO N. BUTTER LDF 11. STF 12. STF 13. STF 14. STF 14. STF 14. STF 15. STF 16. STF 16. STF 16. STF 17. STF 18. STF 18. STF			!	÷	(OHT)++CHHO)	(CH = .18) ÷
* LDF 6487+-R77 11 STF R4,48024-(180) 12 STF R4,48024-(180) 13 STF R2,84024-180 14 STF R2,84024-180 15 STF R2,84024-180 16 STF R2,84024-180 17 STF R2,84024-180 18 STF R2	* :: :: :: :: :: :: :: :: :: :: :: :: ::	8		igo - i		WITEBELY:	
	100 ST	2	3				180 - 247
10	TO 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		2 = TI = R0 +	R1)		DA MACALLITON	1 M - COO . (MI - MA)
R3)	73) :: STF	2	* B + SIN	(R3 = AI + II)		100111101110V	(CG = /GB/ NIS = 10 ·
R3)	R3) :: SIP I PIPPI :: SIP ADDE					94 ++ / XB+	1 NO = SIN , (MY = KZ)
	11. S.	70	1 = 01 - 11	v(1) = B1' = B3)		KZ, ##K3++	
SIF K3, 480.2++	100 H) (Su - 19 - 17)		++4R1,R6,R5	; RO = BI + SIN ; (AR' = R3)
200		8	10 = 10 = 10 = 10 = 10 = 10 = 10 = 10 =			K3, ##22+	;
1 21/20/10			9	50 = 50 - 10 - 10		R1, R0, R2	; (R2 = T1 = R0 + R1)

```
R1, +#R1
                                                                                                                                                                                                       귫
                                                                                                                                                                             • ₽
                             . ë
                                                                                                                                                                                                                                                                                                                                                                                  ; (R2 = TI = R0 - R1)
; R0 = BR + SIN , (R3 = AI + TI)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       44R1++(IRO),R6,R1 ; R1 = B1 + S1M , R3 = 4R + TR
+4RO++,R3,R3
                                                                                     ; R3 = TR = R0 + R5
; R0 = BR + SIN , R2 = AR - TR
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  ; R3 = TR = R0 - R5
; R0 = BR + COS , R2 = AR - TR
                                                                                                                                                                          ; R1 = B1 + COS , (AI' = R4)
                                                                                                                                                                                                                                                                                                                          ; R5 = B1 + COS , (AR' = R3)
                          R2, +480++(1R0), R4 ; (R4 = AI - TI , BI' = R3)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                          ; (R4 = AI - TI , BI' = R3)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               : R3 = AI + II , AR' = R3
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       ; R4 = A1 - T1 , B1' = R3
                                                                                                                                                                                                                                  ; R3 = AR + TR , BR' = R2
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       ; BR' = R2 , (A1' = R4)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  ; R2 = TI = R0 + R1
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  : A1' = R4
                                                                                                                                                                                                                                                                                                                                                                                                               ##R1, R6, R0
R2, ##R0, R3
R2, ##R0++ (1R0), R4
                                                                                                                                                                          AR1++(1R0), R7, R1
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          #+ARO(1),RO
R1,#AR1++(IR1)
#ARO++(IRO)b,R1
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       RZ, +4R3++
R4, +4R2++ (1R0)
R1, R0, RZ
RZ, +4R0, R3
R3, +4R2++
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  4R0++(1R0) b, R1
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    R3, #AR3++(IR0)
                                                       R3, #AR3++ (IR0)
                                                                                                               ##R1++, R6, R0
R3, ##R0, R2
                                                                                                                                                                                                       R4 #4R2++(1R0)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               ##R1++, R7, R0
R3, ##R0, R2
                                                                                                                                                                                                                                     MR0++, R3, R3
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                10, e+4R1(1)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    EINPUT, ARO
EOUTPUT, ARI
EFFTSIZ, RC
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           H-980(1),R0
                                                                                                                                                                                                                                                                                                                          H-4R1,R7,R5
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       R2, +480, R4
R3, +483
R4, +482
                                                                                     70. F3. F3
                                                                                                                                                                                                                                                                  R2.##R3++
                                                                                                                                                                                                                                                                                                                                                       73, #AR2++
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     70, F5, F3
                                                                                                                                                                                                                                                                                                                                                                                  R1. R0. R2
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             2, IRI
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   CLEAR PIPELINE
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     BIT REVENSAL
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             END OF FFT
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          THE SECOND SECON
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                333333
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                ::
BFLEND
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   :: 18 ::
F
```

10, 44481(1)

Appendix A4. Complex, Radix-2 DIT FFT-R2DITB.ASM

The Prince Act of the Prince	APPENDIX AA	•	* THIS PROGNAM INCLUDES FOLLOWING FILES:	
		RZDITB.ASM	* THE TILE THIDINGH. ASM CONSISTS OF THIDDLE FAC	S 30 .
	IC PROGRAM FOR A FAST LOOPED-CODE	RADIX-2 DIT FFT COMPUTATION +	THE THIRDLE FACTORS ARE STORED IN BIT REVERSED • LEDGTH OF N/2 (N = FFTLENGTH).	RDER AND WITH A TABLE
	ON THE TRS320C30	• •	* EXAMPLE: SHOWN FOR N=32, WN(n) = COS(2*Pl*n/N)	- j#SIN(2#PI#n/N)
	en by: Fathund Neyer, Karl Scharf Lehrstur, Fler Nachrichten	EDANIK		
	UNIVERSITAET ERLANGEN-NUER	MEENG +	* 0 R(WN(0)) = COS(2eP1*0/32) = 1	
	CALLERSTRASSE 7, D-8520 EPL	ANGEN, FRG	+ 1 -1(W(0)) = SIN(24P140/32) = 0	
	THEY FX DATA BESTIRE IN INTERNAL	* SHOW IS NOT THE COMPITATION IS NOW.	+ 2	
	ICE, BUT THE RESULT IS MOVED TO A	NOTHER NEMORY SECTION TO +	•	
	STRATE THE BIT-REVERSED ADDRESSIN	* *	*	
	HIS PROGRAM THE MINIMUM FFT LENGT	H IS 32 POINTS BECAUSE OF	+ 13 -1(MN(3)) = SIN(24PI+3/32) = 0.556	
	EPANATE STAGES.	•	+ 14 R(MN(7)) = COS(2+P(+7/32) = 0,195 + 15 -1(MN(7)) = SIN(24P1+7/32) = 0.991	
	TWO PASSES ARE REALIZED AS A FOU	R BUTTERFLY LOOP SINCE THE +	***************************************	
	LIES ARE TRIVIAL. THE MILTIPLIER	IS ONLY USED FOR A LOAD IN	* WHEN GENERATED FOR A FFT LENGTH OF 1024, THE TA	LE IS FOR ALL
	LEL WITH AN ADDF OR SUBF.	•	* AVAILABLE FFT OF LESS OR EQUAL LENGTH.	
		REVERSAL):	+ The MISSING THIRDLE FACTORS (MR(), MR(),) ARR THE STREETIN HANGLAND =	GENERATED BY USING ASILY REALIZED, BY FACTORS AND BY
	. SIZE :	•	•	
		231 WORDS + 512 WORDS + 6	* TO CHANGE THE FFT LENGTH ONLY THE PARAMETERS IN THIDIXBR. ASM AND THE INPUT AND CUTPUT VECTOR LE	The Header of GTHS need to be
FR + J AI	PER BUTTERLY:	•	***************************************	***********
# # # # # # # # # # # # # # # # # # #	STRUES 1 10 8 =		•	
# BR + J BI (COS - J SIN)		, *		, tv ; v; v;
FRR + J BI (COS - J SIN)		10.5 (DUE TO EXT. MEMORY MAITS) +		¥ .
BR + J BI (COS - J SIN) 1 +	SE CYCLES/BUTTERFLY =	* * *	* *	
* 98 + j 81 (COS - j SIN) (+ 1 88 + j 81 (COS - j SIN) (FIN + 11 = 88 + SIN + 81 + COS +		\$272	*	
* 98 + j 81 (0.05 - j SIN) (1.05 - j SIN) (1.1 = 180 + 0.05 + 81 + 6.05 + 81 + 6.05 + 81 + 6.05 + 81 + 6.05 + 81 + 6.05 + 81 + 81 + 6.05 + 81 + 81 + 81 + 81 + 81 + 81 + 81 + 8	LIZATION OVERHEAD = :	2185 = 5.4 % OF TOTAL TIME *	* *	
* TR = BR + CGS + TI = BR + CGS + TI = BR + SIN - SIN	TINE FOR A 1024 POINT FFT =	42 ns (INCLUDING B	(COS - j SIN)	BR/ + j BI/
+ 17 = 189 + 103 + 4 + 16 = 189 + 51N - 13 + 16 = 18 + 17 + 16 = 18 + 17 + 16 = 18 - 18 - 18		REVERSAL) *	· · · · · · · · · · · · · · · · · · ·	
+ AR'= AR + TR + AI = AI - TI + BR'= AR - TI	***************************************	*	NIS # 18 # 200 # 201 # 21 # 4 # 11 # 202 # 12 # 202 # 12 # 202 # 12 # 202 # 12 # 202	
* Al = 18 - 18 - 18 - 18 - 18 - 18 - 18 - 18			- 10 - 11 - 11 - 11 - 11 - 11 - 11 - 11	
			A	
			11 12 12 14 15 15 15 15 15 15 15 15 15 15 15 15 15	
11 + 14 */10 *			N - N - 10 - 10 - 1	

	8	5 P	5 1 2 1 3 1 2 1 3 1 3 1 3 1 3 1 3 1 3 1 3	. 72 - 72 - 73 - 73 - 73 - 73 - 73 - 73 -		. P	2	: R0 = BI + DI , AR' = R0	-	: R1 = BI - DI , BR' = R3		£.	; R1 = C1 , DR′ = R3 = R5 - R1		; R2 = AI + CI , CR' = R2	:	; R6 = AI - CI , DR' = K3	80 + 60 + 70 + 710	21 - 11 - 11 - 11 - 11 - 11 - 11 - 11 -				; KO = CK ; (BI ' = KZ = KZ - KO)	; R1 = BR , (C1' = R3 = R6 + R7)		; R4 = AR + CR , (AI' = R4)	(20 = 78) - 50 = 50		; (DI' = R7 = R6 - R7)	; R6 = DR + BR , (DI' = R7)	20 × 10)		; AR' = R0 = R4 + R6	; R1 = D1 , BR' = R3 = R4 - R6		; KO = BI + DI , AK' = KO	è	; K1 = B1 - D1 , B47 = K3	8 · S	13 + C3 - C	2	: R2 = A1 + C1 . C8' = R2	
w	40 VQV* CQV*	APC, #HRO, K4	MAD 1 400 P.	#681++ #683++ 87	Of De Do	4003++ 4087 R1	R6. R4. R3	R1, #AR1, R0	RO, #4R4±	R1, +4R1++, R1	R3, #AR5++	R1, R5, R2	*+AR2, +AR7, R1	R1,R5,R3	R1, +4R0, R2	R2, #AR2++(IR1)	R1, +AR0++,R6	K3, #986 +	NO, N.E., N.E.	RADIX-4 BUTTERFLY LOOP		BLK1	##2-, ##/, KO	*AR1++, +AR7, R1	R7, R6, R3	RO, #ARO, R4	M4, +9844+	R2. #AR5++	R7,R6,R7	R1, #4R3, R6	R7, ##86++	R3 #482#	R6, R4, R0	*AR3++, *AR7, R1	R6, R4, R3	R1, 44R1, R0	RO, #484+	RI, ##81++, RI	K3, ##(3+	KI, KO, KZ	P+ D5 D2	R1.+400.R2	R2, #M22++(1R1)
FILL PIPELINE	7000				200	in in	1	A SO	i ii	Jens.	STF	ADD	J.	H	a	STE	*	- S	ž	RADIX-4 BUT		8FTB		MPYF.	ADDF	ADD	150	# #S	B 3	400	SIE	! !:	400	HPYF.	3		# ;	B 5	410	ě		A 40	STF
						; INPUT VECTOR LENGTH = 2N (DEPENDS	Carolina in the second control of the second	; UNITUR VELIUR LENGIR = ZM (DEPENDS	(N N)																							OTMINE SAME CANAL	IRO = N/2 = OFFSET BETWEEN INPUTS	, AR7 POINTS TO THIDDLE FACTOR 1	; APO POINTS TO AR	ARI POINTS TO BR	, AR2 POINTS TO CR	t AR3 POINTS TO IR	; ARR POINTS TO ARC	THE DISTRICT OF THE PARTY OF TH	s ARG POINTS TO DRY	HAGNESS OF SET	
TOP N TOP		global MOCHTEL		global SINE		INP, 2048 ; INPUT VECTOR LENGTH = 2N (DEPENDS		UNIT, 2046 ; UNITUI VELIUR LENDIR = 24 (METENDS	2 K5			2	NVIERT-2	WIERT-3	NACHTEL-2	HALB	NHALB-3	_		SIME-1 CINCA	92	1140+2	outh		74		21 + C8′ + C1′	F3: FX + D1	,18	01,	AR7 : FIRST TWIDDLE FACTOR = 1	DETECT.			EINPUT, ARO ; ARO POINTS TO AR	•	•	£	AND OF THE POLITICS TO AND THE POLITICS	••		L, INI. 1. MAINTEN OF MAINTEN OF RA-BUTTEN IES -1. 180 -1. 180 -1. 180	•

	+ FIRST BUTTERRLY-TYPE:	: : : : : : : : : : : : : : : : : : :		200 a 10 a	+ Al' = Al - Ti				RPTB BFLY1		HPYF ++481, 86, 45 ; R5 = B1 + SIN , (487 = R5)	Sent to Sent	**************************************	77, N, 174	***	R3, #4873++		MPYF +4R1+ R6. R0 = BR + SIN R2 = 4R - TR			STF R4, #AR2++	BFLY1 ANDF +ARO++, R3, R5 . R5 = A8 + TR BR' = R2	STF R2, #MR3++	-	+ SALTON ONER TO NEXT GROUP		SUBF R1, R0, R2 ; R2 = T1 = R0 - R1	ADDF R2, #ARO, R3	F5, +402+	* 5	ii Ni	AAA1 - 07 DI - 10 - 10 - 10 - 10 - 10 - 10 - 10 - 1	R4, +4872++(1R1)	#4R1, R6, R0	E S	R0,R1,R3	J.	R3, #480, R2	ADDF +4480++, R3, R5 = 48 + TR , BR' = R2			* SECOND BITTERS V-TYPE:		+ TR = BI + COS - BR + SIN	* T1 = BI + SIN + BR + COS	# AR/ = 48 + TR	+ Al'= Al - TI	NI - 124 1 125 4
A	2		. R1	. C1 ' = 13 = 18 + 187	. Al' = R4 Bl' = R2	!	. D1' = R7 = R6 - R7	; DI' = R7 , CI' = R3									; POINTER TO TAIDBLE FACTOR	: GROUP COUNTER	; UPPER REAL BUTTERFLY INPUT	; UPPER REAL BUTTEDFLY OUTPUT	; LOWER REAL BUTTEBELY OUTPUT	; LOWER REAL BUTTEDFLY INPUT	; DOUBLE GROUP COUNT	; HALF BUTTERFLY COUNT	CLEAR LSB	; HALF STEP FROM UPPER TO LOWER REAL	; PWI	CITO COM OF THE CHANGE TO LOS AND THE	SOLET FRUIT ULLI ITPRILIMENT IU NEM FEME.	DUMPY LOAD DALY FOR ADDRESS LIPHATE	; R7 = COS			AND A LEGISLA MANAGEMENT OF COMMAND	AND - UPTER PERL BUILDINGS & SECTION	; MAI = LUMEN NEAL BUILENFLY INFU!	. AND a linear bear buildings without	THE THE THEFTHEN DEST WE TO COLUMN	: R6 = SIN	: R1 = BI + SIN	; DUMY ADDF FOR COUNTER UPDATE	; R0 = BR + C0S	; R3 = TR = R0 + R1 , R0 = BR + SIN	:	; K1 = B1 + CUS, K2 = AR - TR	\$ \$ \$		
R3, #4866++		¥	22 22	R7.R6.R3	PA. +APA	R2, #485	R7. R6. R7	R7, #4R6	R3, + - #R2		1-2 SIRICE	METER 181	IRO. ARS	1,465	1,486		ESINTAB, AR7	0,484	EINPUT, ARO	ARO, AR2	IRO, ARO, AR3	MR3, MR1	1,486	-2,463	£ .	-1,1%		1,141		#4R1++.R6	##R7, R7			1-					**************************************	#4R1, R6, R1	**************************************	#4R1, R7, R0	+481++,+487,R0	80,R1,R3	P2 400 02	10, 1110, 12 100, 100, 00	R2, #483++	•
:: STF BLK1 ANDF		+ CLEAR PIPELINE	187 5	A	SIF	::		SIF	:: SIF	*		197	ē	SUBI	5		STUFF	Ē	5	<u> </u>		5	5	5 5	5 5	5		5	•	Ē	5	•	GRUPPE	- FILL PIDELING	*				5	JA-JA	:: ADDF	HOYF.			± 7		SIF.	

				-																																£.				Ę	¥		
	S	% - 28	I + BI	= R5 = A1 - B1											8 + 81	8 5	%	R - BI			R + B1	INNER LOOP)	INDER LOOP)	# # #	•					(AI' = R4))		M. = R2)		K, (BC) = F3)	100	. 80 = 88 + COS . (83 = A1 + TL)	!	I, BI' = R3)	,	2 S	, F	= BI + COS . (AI' = R4)	
	. AS = 186 = A	BR' = R7 = AR - BR	; AI' = R4 = AI + BI	.18 ; PS, (9R)	: (48' = F2)	; (BR/ = R3)	; (AI' = RO)	; (BI' = R1)	; A8. = R6	; BR' = R7	; AI' = R4	; B1′ = R5			. AR' = R5 = AR + BI	: AI' = R4 = AI - BR	; BI' = R6 = A	; BR' = R7 = A			; AR' = R3 = AR + B1	; R1 = 0 (FOR INNER LOOP)	; RO = BR (FOR INDER LOOP)	Z	(BR/ = R7)	; (BI' = R6)				; R7 = COS , ((AI' = R4))		; R6 = SIN , (BR′ = R2)	;	; PO = BI + SIN ; (AR' = R3)	(10 + 00 + 11 - 00)	10 + 26 = 0g ·	! !	; (R4 = A1 - T1 , B1' = R3)	1	K3 = K1 = K0 + K0	. K	. R1 = B1 + C0	2
\$	*#RO, +#R1, R6	**************************************	*ARO, *AR1, R4	#AR1++(IRO), +ARO++(IRO), R5 ; B1' = R5 = A1	R2,4482++	R3, #4R3++	RO, #4R2++	R1, #4R3++	R6, +4872++	R7, ##R3++	R4, +AR2++(1R0)	R5, #AR3++(IR0)		*/L	*480++ *+481 65			#4R1++, #4R0++, R7		* / *	*+AR1, *++AR0, R3	+-AR7,R1	+4R1++ R0	##KI++(IKU),+#KO++,KZ ; #K' = KZ = FK = B1 DS #AD244	R7, ##R3++	R6, +AR3++	TERELY:		BF2500	##R7++,R7	R4, +AR2++	•4R7 ++ ,R6	R2, #4R3++	++4R1, R6, R5	13,482±	4481 R7 R0	R2, +480, R3	R2, 4480++(IR0), R4	R3, #MR3++(IR0)	29,57,63 29,53,63	R3 4480 R7	**************************************	D4
2. BUTTERFLY: 💞	ADD	SUBF	ADDA	SUBF	STF	:: STF	STF	:: STF	STF	:: STF		:: SIF		3. BUILDING TY	ADDF	38 78	ADDF			4. BUITERLY: VYI/4	ADDA	5	<u>.</u>	*	:: STF	STF	5. TO M. BUTTERFLY:		eT e	Š	:: STF	Š	SIF.		# S#	Ě	S		:: STF		± 5		
		• 85 = 81 + COS (48' = 85)	!!!!!!!!!!	: (R2 = TI = R0 + R1)	. RO = BB + STM (B3 = A1 + T1)		; (R4 = AI - TI , BI' = R3)	-	; TR = R3 = R5 - R0	; R0 = BR + COS , R2 = AR - TR		; RI = BI + SIN, (AI' = R4)		. 65 = 186 + TR , 1987 = 152				; R2 = TI = R0 + R1	; R3 = AI + TI	G	: DO FOLLOWING 3 INSTRUCTIONS	; R4 = A1 - TI , B1' = R3	:	; R7 = CUS	: BRANCH HEIG			; JUMP OUT AFTER LD(N)-3 STACE				; UPPER INPUT	; UPPER OUTPUT	: LONER INPUT	; LOMER OUTPUT	DISTANCE BETWEEN THE GROUPS						36 + 37 = 23 = 24 ·	
	BFLY2	#+481 R7 R5	PS #482#	R1. R0. R2	08 78 BO	R2. ##0. R3	R2. #480++ R4	R3, ##83++	80,R5.R3	#4R1++,R7,R0	R3, #480, R2	+4R1++,R6,R1	R4, #AR2++	##R0++, R3, R5	nz, menora	<u>w</u>		R1,R0,R2	R2, #480, R3	R5, #4R2++	GRUPPE.	R2, #AR0++(IR1), R4	R3, +4R3++(IR1)	**************************************	##K1++(IR1)		END OF THIS BUTTERFLY GROUP	4, IRO	STUFE	SECOND TO LAST STAGE		EINPUT, ARO	ARO, AR2	IRO, ARO, ARI	ARI, ARI3	5 1R0	eFG8M2, RC		Y	\$ 100 Line 1	2	#4R0 #4R1 R2	THE COLUMN
BI'= AI + TI	er TB	J.	#	ADD	Ž.	9	ABC.	STF	HE ST	HP/F	SUBF	¥.	SIF			CLEAR PIPELINE		ADDA		SIF		JE 35	<u>.</u>	5 5	. <u>a</u>		HIS.	ē	7	5		ē	3	2	9 9	: =	Ē		FILL PIPELINE	į	1	ADD	1

SIF R3, +463		LAST STACE WITH INTEGRATED BIT REVERSAL		LDI ELMPUT, ARO ; UPPER INPUT		EINPUTP2, ARI	edutP1,AR3	esintp2, AR7	BFFTS12, IRO		DI BEGANZ, AC			1. BUTTERFLY: 🗝	ADDF #480, #481, R6 ; AR' = R6 = AR + BR			ADDF +AR1++(IR1), +ARO++(IR1),R5 ; AI' = R5 = AI + BI	Symposis of Constitution of		#F ++4R1,+4R0,R3 ; BR' = R3 = 4R - B1	+-487,R1	*4R1++,R0	#AR1++(IR1), #AR0++, F	1 - 100 + 400 + 4100 = 100 =	R7, #AR2++(IR0)b		3, TO M. BUTTERFLY:			17 CYCLES IF FFT SIZE (1024 DUE TO THE USE OF INTERNAL MEDIORY FOR BIT	REVERSAL, 21 CYCLES IF FFT SIZE = 1024 DUE TO THE USE OF EXTERNAL MEMORY	MEVENDAL.	F +487++ R7 . R7 = CDS ((R1' = R4))	R4. #AR3++(IR0)B		R2, #AR2++(IR0)B		R3, #4R2++(IR0)B	HUNT KI, NV, KZ ; (KZ = II = KV + KI) HDVF + 404 I P7 PA : PA = 10 = KO + KI)	R2. +4R0. R3		R3, #4R3++(IRO)B	MOVF + 42R1++, R6, R0 : R0 = BR + S1N : AR' = R2 = AR + TR	
:: S	S	TS TSP		9	7	9	5	3	3	3	Ē	SMI FIRM LITE		1. BUT	3	ळ	3	₹	TAINT C	3	*	Š		₹ :	± ±5	S FS		3. TO H	PTR		17 CYCL	REVERSA	5	9	SIF SIF		:: STF		:: #S	2	: 53 :::		:: ::	Ē £	
; R5 = AR + TR , BR′ = R2		• PS = RI = SIN (46° = PS)	(CV - VM) 1 10 - 10 - CV -	; (R2 = T1 = R0 - R1)	; R0 = BR + COS , (R3 = AI + TI)		; (R4 = AI - TI , BI' = R3)		; R3 = TR = R0 + R5	; RO = BR + SIN , R2 = AR - TR	200	; Ki = Bi + UD, ; (RI' = N4)	; R3 = AR + TR , BR′ = R2		; R5 = BI + COS , (AR' = R3)		; (R2 = TI = R0 - R1)	; RO = BR + SIN , (R3 = AI + TI)	(Ca - /10 II - [0 = M]) .	(C) = 10 ' 11 - 12 ' 11 '	; R3 = TR = R5 - R0	; R0 = BR + COS , R2 = AR - TR		; RI = BI + SIN , (AI' = R4)	. 55 = 58 + 12 = 58 · = 57	!	200 Aug 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	; ro = B1 + UDS , (AR' = PD)	: (R2 = TI = R0 + R1)	; RO = BR + SIN , (R3 = AI + TI)		; $(R4 = AI - TI , y(L) = BI' = R3)$	8-12-21-22-	; R0 = BR + C0S , R2 = AR - TR		; R1 = B1 + SIN , R3 = AR + TR				: BR/ = R2 . A1/ = R4		; R2 = TI = R0 + R1	: K3 = K1 + 11 , EK	; R4 = AI - TI , BI' = R3	
	R2, ##R3++	F+481.86.85	_	_					_		_	R4 #AR2++	_	R2, #4R3++	++4R1,R7,R5	_	_		K2, +860, K3	_	_			PARI++, K6, R1		_		F + 682 + CB	_			K2, +400++, K4			_		#4R0++,R3,R3			R2, +483++				R2, #AR0, R4	
ADD	±s -	J.A.	STF.		HPYF	:: A004		::	A POC			::		:: STF	HANE	:: SIF	18 7			HS STE	187 5			::		:: STF	•	:: ::				# E				BF 2500 HPYF		• •	יי עניא יוידנוות *	STF	:: SIF	ADDF	# #S		

```
; (R2 = TI = R0 - R1)
; R0 = BR + SIN , (AI' = R3 = AI - TI)
                                                                                                                                                                      ; R3 = TR = R0 - R5
; R0 = BR + COS , AR' = R2 = AR + TR
                                                                                                                                                                                                             ##R1++(IR1), R6, R1 ; R1 = B1 + SIN , BR' = R3 = AR - TR
                                                                                                                                           ; (BI' = R4 = AI + TI , AI' = R3)
                                                                                                                                                                                                                                                                                                    ; R2 = TI = R0 + R1
; AI' = R3 = AI - TI , BR' = R3
                                                                                                                                                                                                                                                                                                                                            ; BI' = R4 = AI + II , AI' = R3
                                     ; BR' = R3 = AR - TR , AR' = R2
R3,4460,R2

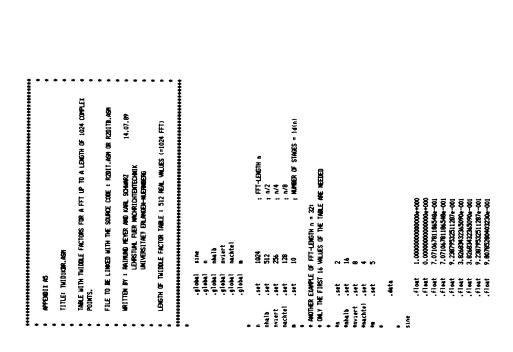
48R1-4(R1),R7,R1 ; R1 = B1 + COS , (B1 '= R4)

R3,4463-4-(180)B

R2,4462-4-(180)B

R2,4462-4-(180)B
                                                                           ; R5 = BI + COS , (BR' = R3)
                                                                                                                                                                                                                                                                       ; AR' = R2 , (BI' = R4)
                                                                                                                                                                                                                                                                                                                                                                    ; BI' = R4
                                                                                                                   +4R1, R6, R0
R2, +4R0, R3
R2, +4R0++ (IR1), R4
R3, +4R3++ (IR0) B
                                                                                                                                                                                                                                                                            R2, +4R2+(1R0) B
R4, +4R3++(1R0) B
R1, R0, R2
R2, +4R0, R3
R2, +4R0, R4
R3, +4R3+(1R0) B
R4, +4R3
                                                                                          R3, #AR2++(IR0)B
                                                                                                                                                                                   #4R1++,R7,R0
                                                                                                                                                                                                                         R3, #ARO++, R3
                                                                             ***AR1,R7,R5
                                                                                                                                                                                                R3, #4R0, R2
                                                                                                                                                                       RO, R5, R3
                                                                                                       R1, R0, R2
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         滋
                                                                                                                                                                                                                                                   CLEAR PIPELINE
                                                                                                                                                                                                                                                                                                                                                                                                END OF FFT
                                                                                                                                                                                                Ą
                                                                                                                                                                                                                                                                             STF ADDIT
                                                                                                                                                                                                                                                                                                                                                                                                                          $ $ $ $
                                                                                                                                                                                                ::
BFLEND
                                                                                                                                                                                                                                                                                                                                                                                                                                                                               ₹
                                                                                                                                                                                                                          :: .
                                                                                                                                 ::
                                                                                                                                                          ::
                                                                                                                                                                                                                                                                                                                                                          ::
```

Appendix A5. TWID1KBR.ASM—Table With Twiddle Factors for a FFT up to a Length of 1024 Complex Points.



7.11432195745216e-001 7.02754744457255e-001 6.13589444915452e-003 9.99981175282601e-001

Appendix B. Radix-4 Complex FFT

Appendix B1. Generic Program to Do a Looped-Code Radix-4 FFT on the TMS320C30

UPONT, 1 ; SECOND-LODP COUNT 01, 1 ; UT COUNTER IN PROGRAM, P. 117 IA1, 1 ; IA1 INDEX IN PROGRAM, P. 117	; INITIALIZE DATA LOCATIONS	TEMP ; COMMAND TO LOND DATA PAGE POINTER ETEMP, ARO ; COMMAND TO LOND DATA PAGE POINTER ESTURE, ARI	##O++, RO ; XFEN DATA FROM ONE NEDWAY TO THE RO, 4461++ RO RO, 4461++ RO RO, 4461++ RO RO, 4461++ RO	196+ 'QI 94 '096+ 196 '096+ 197 '197 '197 '197 '197 '197 '197 '197 '	FFTSIZ ; COMPAND TO LOND DATA PAGE POINTER BFFTSIZ, RO BFFTSIZ, IRO BFFTSIZ, IRI	ARTY ESTAGE ; ESTAGE HOLDS THE CLARENT STAGE : NUMBER		AR7, EMPTONT ; INITIALIZE REPEAT COUNTER OF FIRST ; LOOP	-2,R0 AR7,eVEENDX ; INITIALIZE IE INDEX 2,R0 R0,&JT ; JT=R0/2>2	2,R0 1,R0 ; RO∰Z		EINPUT, 480 ; 480 POLNTS TO X(1) RO 480, 481 ; 481 POLNTS TO X(11) RO 481, 482 ; 482 POLNTS TO X(12) RO 482, 483 ; 483 POLNTS TO X(13)	1, PC ; PC SHOLLD BE ONE LESS THAN DESINED # BLK1 ##80, ##82, R1 ; R1=V(1)+V(12) ##80, ##81, R3 ; R3=V(1)+V(13) R3, R1, R6 ; R6=R1+R3
88.88	• FF •	9 5 5	•		3555	i ii.	5.5 5.	STI *	LSF STI A001 STI	1805	* OUTER LOOP	L00P: A001 A001 A001 A001 A001 L01	9.081 • F1ST LOOP • RPTB AUDE AUDE
	GENERIC PROGRAM TO DO A LODPED-CODE RADIX-4 FFT COMPUTATION ON THE TRS20C30.	THE PROGRAM IS TAKEN FROM THE BURBLIS AND PARKS BODD, P. 117. THE COMPLEX DATA RESIDE IN INTERMAL REDIGNY, AND THE COMPUTATION IS DONE INH-FLACE.	THE THIRDLE FACTORS ARE SUPPLIED IN A TOME PUT IN A .TOMTA SECTION, THIS DATA IS INCLUED IN A SEPARATE FILE. TO PRESENCE THE CREATION WILDER OF THE PROGRAM, FOR THE SAME PARPOSE, THE SIZE OF THE FIT IN AND LOCA(IN) ARE DEFINED IN A .GLOBL DIRECTIVE AND SPECIFIED DURING LINKING.	IN ODGER TO HAVE THE FINAL RESULT IN BIT-REVERSED DOGER, THE THO MIDDLE BRANCES OF THE RADIL-4 BITTERELY ARE INTERCHANCED DURING STORAGE, NOTE "HIS DIFFERENCE MEN COMPARING WITH THE PROCESAN IN P. 117 OF THE BURBUS AND PARCE SHOCK."	* AUTHOR: PRANCS E. PRPANTCHALIS * AUGUST 23, 1987	; ENTRY POINT FOR EXECUTION	; FFT SIZE ; LOG4(N) ; ADDRESS OF SINE TABLE	.024 ; MEMORY WITH INPUT DATA		; STARTING LOCATION OF THE PROGRAM	; RESERVE 100 WORDS FOR VECTORS, ETC.	; BEGINNING OF TEMP STORAGE AREA	1 ; FFT S12E 1 ; LOG4(FFTS12) 1 ; SINCOSINE TABLE BASE 1 ; ARECA WITH INPUT DATA TO PROCESS 1 ; FFT STAGE # 1 ; REPEAT COUNTER 1 ; IE JNUEX FOR SINE/COSINE
IX B1	PROGRAM TO DO 30.	GRAM IS TAKEN I SIDE IN INTERN	DOLE FACTORS AN INCLUDED IN A INCLUDED IN A INCLUDEN IN A INCREMENTAL DISTRICT OF THE SAME IN A INCREMENTAL	IN ORDER TO HAVE THE I BRANCHES OF THE RADIX: THIS DIFFERENCE LANEN (AND PARKS BOOK, ,	AUTHOR: PANOS E. PAPAMICHALIS TEXAS INSTRUMENTS	.0L08L FFT	.GL08L N .GL08L N .GL08L SINE	.USECT "IN", 1024	.TEXT	.WORD FFT	SPACE 100	MORD 6+2 MORD FFTS12 MORD N MORD SINE MORD INP	S LOGFF1,1 S SINTAB,1 INPUT,1 S STAGE,1 S PRTONT,1 S RETONT,1
* APPENDIX B1	* GENERIC PR * THS320C30.	+ THE PROC + DATA RES	THE THILL DATA IS PROGRAM. PROGRAM.	* IN ORDER * BRANCHES * THIS DIF * AND PARK	AUTHOR:	ક	दंदंदं	INP.	TEXT INITIALIZE	9. • •	· •	TEPPWORD STOREWORD WORD WORD WORD	* ************************************

LDI etat,487	ADDI ESINTAB, ARA ; CREATE COSINE INDEX ARA		SUBI 1, ARS ; 1A2=1A1+1A1-1	APS, ARS, AR6	SUBI 1, AR6 ; IA2=1A1-1	•	4 SECOND FOOD	-	BUK2	••	ADDF *+AR3, *+AR1, R5 ; R5=Y(11)+Y(13)	ADDF R5, R3, R6 ; R6=R3+R5	SUBF #+482, ++480,R4 ; R4=Y(1)-Y(12)	•-	+AR2, +AR0, R1	*AR3, *AR1, R5 ;	MPYF R3, #+AR5(IR1), R6 ;	R6, *+AR0	R5, R1, R7	#4R2, #4R0, R2		MPYF RI, +4R5,R7		2004 K/, K6 ; NO-MORTON (17) (17) (17)	81 ++4R5(1R1) R7 ·	B4 ****	MPYF R3. #M5. R6	R7, R6	R5, R2, R1	23,53	*44R3, *44R1, R5	R5,R4,R3		CTT K3, #**#K4 (JK1), K0	CONTINUE ON	R7 R6	R1, #+4R4(1R1), R6	R6, *+4R2	R3, +AR4, R7	R7,R6	MPYF R4, *+AR6 (IR1), R6 ;	R6, +AR2++(IR0)	R2, #486, R7	R7, R6	MPYF R2, #+AR6 (IR1), R6 ;	R6, ++4R3	R4, #486, R7	ADDF R7, R6	BLK2 STF R6, +AR3++(IR0) ; X(I3)=R2+CU3+R4+5I3	-	
; R4=Y(I)-Y(I2)	; Y(I)=RI+R3	; RI=RI-R3	; PS=X(12)	; R7=Y(I1)	; R3=X(11)+X(13)	; R1=X(1)+X(12)	; Y(II)=KI-K3	; R6=R1+R3	; R2=X(I)-X(I2)	; X(I)=RI+R3	. R1481 43	; R6=X(11)-X(13)	; -R3=V(11)-V(13) !!!	: X(II)#(II)X :	: PS=R4-P6	: R4=84+R6	; Y(12)=R4-R6	; Y(I3) =R4+R6	: 155±157±153	; R2#2#3	; x(12) = R2 - R3	; X(13)=K2+K3							; CURRENT FFT STAGE					; INIT IAI INDEX	COO - Charl Con Courses Coo - Tier	; INTI FOUR COUNTER FOR TIMES FOUR	. INCREMENT INNER LOOP COUNTER				; IAI=IAI+IE	; (X(I),Y(I)) POINTER		; (X(II),Y(II)) POINTER		; (X(12), Y(12)) POINTER	; (X(13),Y(13)) POINTER		; RC SHOULD BE ONE LESS THAN DESIRED #	; IF LPONT=JT, 60 T0	; SPECIAL BUTTERFLY
*+AR2, *+AR0, R4	R6, ++AR0	R3,R1	#4R2, R5	*+AR1,R7	#AR3, #AR1, R3	F5, ##80, R1	RI, ++4RI	R3,R1,R6	R5, +4R0, R2	R6, #4R0++ (IR0)	13,81	#AR3, #AR1, R6	R7, 4+4R3, R3	RI, #4RI++(IRO)	R6, R4, R5	R6,R4	F5, ++4R2	R4, ++AK3	R3,R2,R5	23,62	R5, +AR2++(IR0)	R2, #AR3++(IR0)	THE LACT CTACE WAS ABO	IF INIS IS THE LAST STANE, THE BURE	PSTACE AR7	1 487	PLOGFFT, AR7	93	AR7, ESTAGE		100b		1,487	PK/, EIA1	2,187	W/ Brown	2.AR6	BLPONT, ARG	ELPONT, ARO	EIA1,AR7	EIEINDX, AR7	EINPUT, ARO	AR7, EIA1	RO, ARO, ARI	AR6, ELPONT	RO, AR1, AR2	RO, AR2, AR3	ERPTONT, RC		€JT,AR6	2
38 8	STF	HIS.	_	<u>+</u>	A004		::	A004	SUBC	:: SIF	308	SUBS	3000	:: SIF	SUBC	A00 5		SIE			=	SIF	* .	SI SINI AI	107	; ;	Ideo	028	STI	•	* MAIN INNER LOOP	•	5 i	115	∄ 8	116 (9)	197	400	ē	ij	100	4001	STI	700 7	STI	400 1	AD01	9	igns i	E	078

STI PAGE, ELEINDX . MILLIADY	-3,80	2,780	511 NO. W. 1	4/2H-2N: 02'1 35'		STORE RESULT OUT JEINS RIT-REGEREN AMORESCING	THE PROPERTY OF THE PROPERTY O	_	1,80			t time(), into			RPTB BITRV			FO, ++#R1(1)			SELF ; BRONCH TO ITSELF AT THE END	9																							
12 =	3	₹ 8	⊼ 7	ુ	£	STORE		107 :043	3	3	5 5	<u> </u>	9		82	Ė		⋧	STF		86 i	3																							
; LOOP BACK TO THE JIMER LOOP					; PUINT TO SINC(45) ; CREATE COSINE INDEX ARK=COZI			; R1=X(1)+X(12)	; KZ=X(1)-X(12)	: Dar(1)+(12)	: R5=K(II)+X(I3)	: R6-R5-R1	; R1 +R1 +R5	; R5=Y(II)+Y(I3)	; R7=R3-R5	; K2+K2+K5	5. Y(1) + 13. H(2) + 1	: X(I)#(I+E)	; KI=X(II)-X(I3)	; K#*(11)-*(13)		; 1(1) HC+C3+C3	:	: 153m64-R1	: R4=R4+R1	: R1=R3-R5	; RI=RI+6021	: K3#K3+5	; K3#G#UZ1	; T(12)=(K3-K3)+(J)1	. RI-SI-M21	; X(12)=(R3+R5)+C021	; R2=R2+R4 !!!	; R2=R2+0021	; Y(13)=-(R4-H2)+C021	; K(13)=(K4+KZ)+UJZ1			; LOOP BACK TO THE INNER LOOP			; Increment repeat counter for Next	#11 :	3144=31 :	
BLPONT, RO INLOP	SONI	SPECIAL BUTTERFLY FOR HEJ		IR1, AR4	ESINTAB, ARA		E C3	#4R2, #4R0, R1	**************************************	**************************************	**************************************	R1, R5, R6	72, 23 124	#+AR3, #+AR1, R5	P5, R3, R7	5. 5.	K3, #+490	K1, #9K0++(1K0)	11, 11, 11, 11, 11, 11, 11, 11, 11, 11,	DA AADI	D7 #AD144/TD01	17, marit 11(1)	2,5,2	R1, R4, R3	R1, R4	R5, R3, R1	**AR4,R1	S, 6	**************************************	PA PO PI	+AR4 R1	R3, #AR2++(IR0)	R4, R2	+AR4, R2	K1, #48K3	NZ, THEKSTY (INU)		BLPCMT, RO	INLOP	FOOT INCLUDE	@IEINDX, AR6	2,487	TWO DESTRUCTIONS	2,486	
ā as	%	PECIAL BUT		<u>5</u> 5	S S		æ		8		A P	B	ADD	9			<u>+</u>	100	500	ž į	5 5	Anne	2	B	A 004	#BS	J. A		Ė	1 de 7	¥.	STF	ADD	d t	<u>+</u> =	LIC.		i d	6	ē	ë	<u>*</u>	115	3	i
	•	•	•	ಕ್ಷ		•											=	Ξ			=	=							=	:		::		2	ž :	: .	•			. E			•		

Appendix B2. fft_4-Radix-4 Complex FFT to Be Called as a C Function

	; ENTRY POINT FOR EXECUTION ; ADDRESS OF SINE TABLE				RO ; NOVE ARGUMENTS TO LOCATIONS MATCHING S12 ; THE NAMES IN THE PROGRAM FPT RO FTT RO FTT TO FTT TO	FT STROKE # FEPS FEPS		IN WHERE IN WATER IN THE WAY TO THE WAY IN T
PR3	SINE	FFTSIZ, 1 LOGFT, 1 IMPUT, 1	C FUNCTION		LDI ++P(2), RO STI RO, GETSIZI LDI ++P(3), RO LDI ++P(4), RO STI RO, ELLOSFT STI RO, ELLOST STI RO, ELLOST INITIALIZE FFT ROUTINE	STAGE, 1 RPTCNT, 1 IEINUX, 1 IPONT, 1 JT, 1 IA1, 1	6FTS12, RO 6FTS12, IRO 6FTS12, IR1 0, AR7 AR7, ESTAGE	1, IR0 -2, IR1 1, AR7
* FP .SET	18020. 18020.	* 883.	E	tift.it. Post DI DI Post Post Post Post Post Post Post Post	. INTIALISE	88. 88. 88. 88. 88. 88.	•	क के ज
•	TO BE CALLED AS A C FUNCTION.	SISS. Int fft_4(N, N, DATA) Int N FFT SIZE: N=4++1 Int N NUMBER OF SIABES = LOG4(N) Float +6-4ta APRRY NITH INFUT AND CUTPUT DATA IPTION:	THE DATA 4880Y IS 244-10MG, WITH REAL AND INACINARY WALLES BLTER-MATING. THE PROCREM IS BRIGED ON THE FORTRAM PROCREM IN THE BURBUS AND PAGES BODK, P. 177.	IN ORGEN TO HAVE THE FINAL RESULT IN BIT-REVERSED ORDER. THE THO NIDDE BROADCES OF THE RADILY-4 BUTTERFOR WE INTECHMED DRING NIDDE. BROADCES OF THE RADILY-4 BUTTERFOR ORDERING WITH THE ROOMSHOW ON P. 117. THE COPPUTATION IS DONE IN-PLACE, AND THE ORIGINAL DATA IS DESTRUCED. BIT REVENERS AT INTELPRATED AT THE BUD OF THE PROTITON. IN THIS NOT NECESSARY, THIS PART CAN BE CONDITION OUT. THE DIRECTION OF STANDARD STANDARD STANDARD TO BE SUPPLIED DURING LINK TIME, AND IT SHOULD HAVE THE FOLLOWING FORMAT:	.sine .float value1 = sin(0628pi/N) .float value2 = sin(1528pi/N) .float value(5M/4) = sin(1504/4-1)#28pi/N) THE VALUES value1, value2, ETC., ARE THE SINE WAYLES. FOR AN H-POINT FIT, THERE GRE MANA, VALUES FOR AN CANE OF		15, 16, 17, 180, 181, 182, 183, 1844,	MAS, ARA, AR7, IRA, IRI, RS, RE, RC PREMICHALIS STRUMENTS COTOBER 13, 1987
APPENDIX B2	NAME: FFL.4 RADIX-4 COPPLEX FFT TO BE CALLED AS A C FUNCTION.	SYMPSIS: INT N FFT SIZE: N=4+++ INT N FFT SIZE: N=4+++ INT N NAMES OF SIAES = LOG4(N) float 444ta APRRY NITH INPUT AND OUTPU DESCRIPTION: GENERIC FUNCTION TO DO A RADIX-4 FFT COPPU	THE DATA APROV IS 244-10MG, MITH MATING, THE PROGRAM IS BASED ON AND PAPERS BOOK, P. 117.	THE STATE OF THE THAN ESSLY IN BIT-REPESSO ORDER, THE THE ORDER OF THE RADIAL PAIRTENANCED ORDER. THE THE STATE OF THE STATE OF BUTTERFOR HE DISTRIBUTED OF THE PROGRAM STORGAE. NOT THIS DIFFERENCE WERE CHARMEN WITH THE PROGRAM P. 117. THE COMPUTATION IS DONE IN-PLACE, AND THE ORD OF THE PROGRAM IS THIS THIS IS NOT NECESSARY, THIS PART CAN BE CONCENTED OUT, THE STATE OF THE PROGRAME OF THE PROGRAME OF THE PROGRAM STATE OF THE PROGRAM ST	data Jane float valuel = 51 float valuel = 52 float value(30/4) THE WALES valuel, value2, ETC. H POINT FT INEER RE HAVE AND FE THE GAME IN THIS LAW.	STACK STRUCTURE UPON THE CALL: -FP(4) DATA -FP(3) N	#P(2) N N FETURA ALOR : FP(1) RETURA ALOR : FP(1) OLD FP : FP(1) FP FP FP FP FP FP FP F	ARS, MR, AR7, 180, AUTHOR: PANUS E. PAPONICHALIS TEXAS INSTRUMENTS

	, co. ,	1,447) 807 8161 . THIT TAY TANCK		AR7, BLPCHT ; INIT LOOP COUNTER FOR INNER LOOP	2 Ab.	767	E-POT-ABO	EIAL AR7	elEINDX,AR7 : IA1=IA1+IE			RO, ARO, ARI ; (X(II), Y(II)) POINTER	PAGE ADDITION AND AND AND AND AND AND AND AND AND AN		BRPIONT RC (ATLS), TILS) FUINIER		€JT, AR6 ; IF LPCNT=JT, GO TO		EIA1, AR7	EIA1, AR4	ESINTAB, AR4 ; CREATE COSINE INDEX AR4		1,ARS ; IA2=IA1+IA1-1	5, A66	1,AR6 ; IA3=IA2+IA1-1			BELK2	**AR2, **AR0, R3 ; R3=Y(1)+Y(12)	 82.13	••	#+48KO, R4	RO, K3 ; R3#R3-R5	###X,###O,KI ; KI=KII/#KIIZ)	 ă	٠.	•	24		5,87	•-	••	••	IR1),R7 ;	••	K3, *MCJ, K6 ; K6-MCJ, K6 ; K6	//, N6 ; R6=R1 #C02+R3#S12
+ MAIN INNER LOOP	•	5	: 5	118	IMCOL.	1000	3	Ē	4001	A001	STI	1007	SII	1000	į	IBNS	I d O	OZ8	ij	ë	ADDI	ADDI	Igns	A001	ians	a001 000355 +	*	RPT8	ADDF	A00F	500	#878 8				315	ADDE					_		14.0	100	_	_
; INITIALIZE IE INDEX	; JT=R0/2+2	!					AND POINTS TO X(I)	AND POINTS TO X(11)	ADD BOTHER TO VITAL	THE LOWIS TO ALLS!	: PC SHOULD BE ONE LESS THAN DESIRED #					(Z))+(I)+(I)+(I)	. 10/-11/11/11/19/	. D4=V(1)-V(12)	(71)1-1(1)	111/#1#3 101/6-103	(1) A=101	: RZ=V(11)	: K3=X(11)+X(13)	: R1=X(1)+X(12)	; Y(II)=RI-R3	; R6=R1+R3	; R2=X(I)-X(I2)	; X(I)=RI+R3	: KI-KI-K3	; KO=X(11)-X(13) ; -P2=V(11)-V(13)		: P5=R4-P6	; R4=R4+R6	; Y(12)=R4-R6	; Y(I3)=R4+R6	: R5-R2-R3	; R2=R2+R3 !!!	; X(12)=42-R3	; X(13)=R2+R3 !!!							: CURRENT FFT STACE	
-2, RO AR7, @IEINDX 2 RO	R0, €.T	9,30	J. w			And House	00 00 00 00 00 00 00 00 00 00 00 00 00	100, OBC 1000	80 082 083	BRPTONT RC	1,80				BLK!	##007 ##02, KI	R3 R1 R4	#+000 #+000 B4	Dr **ADO		90 00	*+AR1.R7	#4R3, #4R1, R3	R5, #4R0, R1	R1, ++4R1	R3,R1,R6	72, +480, F2	R6, #4R0++(IR0)	K3, K1	R7 #+483 R3	R1, #4R1++(1R0)	R6, R4, R5	R6, R4	P5, *+4R2	R4, ***#R3	73,72,75	R3,R2	F5, #M2++(IR0)	KZ, ##K3++(IRO)	THIS IS THE LAST STATE WAY ARE PROME	E CASI SIRACE, TOURE	PSTAGE AR7	1.487	6L0GFFT, AR7	96	AR7, eSTAGE	
75 ES	Its	3 3	•	* OUTER LOOP	•		3 5	§	ADDI	Ē	IBNS	•	+ FIST LOOP	•	S S S S S S S S S S S S S S S S S S S		ADD		#	1	<u> </u>	ë ::		ADDF	:: STF	P		15			:: SIF	SUBC	ADDF		±S ∷	*		# F	<u>.</u>	- 15 THIS IS TO		igi	ADDI	Idlo .	020	STI	

2.056 4683, 4681, 185 400 50 483 400 50 60 483 400 50 60 60 60 60 60 60 60 60 60 60 60 60 60	RE-A(11) - K(13) Re-A(11) - K(13) - K(13) Re-A(11) - K(13) - K(13) - K(13) Re-A(11) - K(13)	:: :: 38 :: • • 00 •		FG, RG R4, RR R4, RR R4, RR R4, RR R4, RR R4, RR R5, 4464, RR R6, RC R4, RC R4, RC R4, RC R6,	R946346 R94634021 K122=K8-46594021 K123=K8-469 4021 K123=K8-469 4021 K123=K8-469 4021 K133=K8-462 40221 K133=K8-462 40221 K133=K8-462 40221 K133=K8-462 40221 K13462 4047 TT
	RG-444-155 RG-441-155 RG-476-151 X (III) = RI + 0020-RG-85 2 X (III) = RI + 0020-RG-85 2 X (III) = RI + 0020-RG-85 1 RG-478-1001 RG-			######################################	KP-RE-MOZZ KP-RE-MOZZ KI (12) = (RE-MS) + (MZ) + (
	Re-Reference Reference			R1, 4-1487, R4, R4, R4, R4, R4, R4, R4, R4, R4, R4	MITCH MITC
	Redeficition Redeficition Redeficition Redeficition X(III)=R[action + Redeficition X(III)=R[action + Redeficition			R4, F2, F1 	11.00 BACK TO THE THACK TO THACK TO THE THACK TO TH
	KH-AFR-6001 KH-AFR-6001 RF-400-601 RF-400-601 RF-400-601 RF-400-601 RF-400-601 RF-400-601 R			## RZ, ##	R R R R R R R R R R
	(x (11)1-RE-1002-RE-1512 RO-410-011 RO-410-011 RO-410-011 (x (12)-450-RE-110 RO-410-011 RO-410-011 RO-410-01-RE-110 RO-410-01-RE-111 RO-410-RE-111			######################################	RI-48 (2021) KI (12=(R2-4R3)+0021 RO-4C2-4R4 (11) RO-4C2-4R2 (11) KI (33=(R4-4C)+00221 (11) KI (33=(R4-4C)+00221 (11) TIPE IE-4-1E MI-4C2 JI-4C2-42
	RA-RESTILL		517 517 517 518 518 519 519 519 511 511 511 511 511	RR, 4R2 + 4R2 + 4R4 + R2 + 4R4 + R4 +	KI(12)=RAND) + MOZ ROPENDO 111 KI(13)=RA+R2)+GO21 111 KI(13)=RA+R2)+GO21 111 LOP BACK TO THE INNER FOR MEXT INCREMENT REPEAT COUNTER FOR MEXT ITHE IE=4=1E MI=4/2 JF=4/2/2-2
	R6-48-001 -R1+511 R6-48-001 V1(12) +R6-48-001 R7-48-061 R6-48-001 -R2-511 R7-48-061 R6-48-000 -R2-513 R6-48-000 -R2-513 R6-48-000 -R2-513 R6-48-000 -R2-513 R7-48-613		STATE OF THE STATE	RM, R2 +884, R2 R1, +4481, R2 R1, +4482 R2, +483+(1R0) R2, +487 BETONT, R6 BETONT, R6 BETONT, R6 R7, 886 R8, 81E1MIX R0, 1R0 -3, R0 -3, R0 -5,	R-RC-R01
	VILT2 #58-01-01-51 VILT2 #58-01-01-51 R7-#0-50 R7-#0-50 VILT2 #58-01-47-51 R6-#1-001-47-51 R7-#0-50 VILT3 #58-001-47-50 R7-#0-50 VILT3 #58-001-47-50 R7-#0-50 VILT3 #58-001-47-50 VILT3 #58-001		ST S	+ + + + + + + + + + + + + + + + + + +	RP-RP-RD21 11
	(* (12) #(3+00) = (3+0		ST THE ST	R1,++483 (Z2,+4823++1180) (BC,2) (BC) (BC) (BC) (BC) (BC) (BC) (BC) (BC	; V(13)=-(84-62)+0021 !!! ; V(13)=-(84-62)+0021 !!! ; LOOP BACK TO THE INNER LOOP ; TITNE ; IE=4+1E ; NI=4/2 ; JI=4/2/2+2
	N7-40-5-11 N7-40-5-11 N4-40-40-01-40-5-11 N112-3-40-01-42-5-11 N4-40-40-40-42-5-13 N4-40-40-40-42-5-13 N7-40-41-3 N7-40-4 N7-40-4 N7-40-4 N7-40-4 N7-40-4 N7-40-4 N7-40-4 N7-40-4 N7-		1	R2.;48(3++1R0) 18.00 18.00 18.00 2.48(7) 48(7),48(8) 2.48(7) 48(7),48(8) 48(8)	; X(13)=1844R2)+6021 !!; ; LOOP BACK TO THE INNER FOR NEXT ; THE ; IE-4+1E ; M1-4/2 ; J1-4/2/2-2
	M-47-40-11			ELPONT, RO INLOP BERTONT, AR7 ELEINDY, AR8 2, AR7 2, AR7 RO, IRO RO, IRO RO, ELEINDY RO, IRO RO, ELEINDY RO, ELEIN	LOOP BACK TO THE INNER LOOP
	Re-Re-Re-SI	00 .	1968 1971 1971 1971 1971 1971 1971 1971 197	BLPOHT, RO INLOP BEETONT, ARA BEETONT, ARA ARAY, BERTONT ARAY, BRAS ARAS, BEETONT RO, IRO -3, RO -3, RO -5,	; LOOP BACK TO THE INNER LOOP ; INCREMENT REPEAT COUNTER FOR NEXT ; ITHE ; IE-441E ; MI-MC ; JI-42/242
	Kill	• 00 •	1868 1971 1888 1888 1888 1888 1888 1888 188	INLOP INLOP GENTONI, AND GENTONI, AND Z, AND Z, AND AND, GENTONI AND, INCO Z, NO Z,	; LOOP BACK TO THE INNER LOOP ; INCIDENT REPEAT COUNTER FOR NEXT ; TITE ; IE=401E ; NI=402 ; JT=402.2×2
	X(12)=40+60+60+60+60+60+60+60+60+60+60+60+60+60	• 00MI	04. 101. 15. 15. 15. 15. 15. 15. 15. 15. 15. 1	INLOP BERTONT, AR7 BERTONT, AR8 2, AR7 487, BERTONT 2, 486 R0, 1R0 3, R0 3, R0 5, R0 1, R0	: LODP BMCX TO THE INNER LODP : INDOEHENT REPEAT COUNTER FOR NEXT : ITINE : IE-401E : M1-402 : JT-402/242
	R7-#C2-613 R4-#46-003-R2-813 R6-#R2-003 V.123-#R4-603-R2-813 R6-#C2-003-R4-813 X.1131-#C2-003-R4-813	• 00M	100 H 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	BEPTONT, 487 2, A87 2, A87 2, 886 2, 886 487, BEPTONT 2, 886 2, 886 486, EIENBOX 1, 180 2, 800 1, 180 1, 180 1, 180 1, 180 1, 180 1, 180 1, 180	INCREMENT REPEAT COUNTER FOR MEXT TIME TEMPLE TEMPLE MINING MIN
	R6-46+003-R2+513 R6-470+003 Y1(3)=R4+003-R2+513 R7-40+613 R1(13)=R2+003-R4+513 X1(13)=R2+003-R4+513	• COM1	1987 23 11 1897 23 11 187 187 187 187 187 187 187 187 187	eertont, 487 2, 487 2, 487 2, 486 487 2, 486, 8EEHOX 480, 8EEHOX 5, 780 2, 780 60, 180 2, 780 2, 780 1, 80	; INCREMENT REPEAT COUNTRY FOR NEXT ; THE ; IE-441E ; NI-H2 ; UT-H2/242
	Memorano	<u>.</u>		2,487 2,487 487,489TCHT 486,81E1MEX 496,81E1MEX 13,780 2,3,780 190,801 190,801	; INCREMIT REPEAT COUNTER FOR NEXT ; TIME ; IE-4-1E ; NI-4/2 ; JI-4/2/2-2
	VII3 ##################################			2,1427,1420,1420,1420,1420,1420,1420,1420,1420	; INDEPENDENCE FOR NEXT ; TIME ; IE=44 IE ; NI=42 ; JT=42/2+2
	VI 33-440-003-42-42.33 FR-470-003-44-45.33 K(13)-472-403-44-45.33			2, NeV / May	
	; R7-#4e513 ; R6-#2e039-#4e513 ; X(13)-#22e039-#4e513	•	ST 15.24 ST	AR7, ##PPTCMT 2, AR6 AR6, #LE1NDX R0, 1R0 -3, R0 2, R0 R0, &JT R0, &JT R0, &JT R0 1. R0	: 1146 : 11462 : 1746242
	; X(I3)=MZ#CXX+NA+SI3 ; X(I3)=MZ#CXX+NA+SI3		ST 15-15-15-15-15-15-15-15-15-15-15-15-15-1	AR7, 48P TOHT 2, AR6, AR6, EIEINOX AR6, EIEINOX -3, R0 2, R0 R0, eJT R0, eJT -80	; IE=44E ; MI=42 ; JF42/242
	; X(I3)=R2e034R4eSI3		15 C C C C C C C C C C C C C C C C C C C	2,486 486,e1E1NDX RO,1RO -3,RO 2,RO 2,RO 1.RO	314=31 : 247/28=10
			ST 1 100 A 2001 1 1 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	AR6, ELEINDX R0, IR0 -3, R0 2, R0 20, E-1T 2, R0 1. R0	; NI=RZ ; JF=RZ-Z-Y-Z
			1973 1983 1983 1983 1983 1983 1983 1983 198	RO, 1RO -3, RO 2, RO RO, EJT 2, RO 1, RO	24/20+10 ;
			19 15 15 15 15 15 15 15 15 15 15 15 15 15	78, 180 2, 80 80, &JT 1.80	5-2/2)#10 :
21 EC 121 180			15. S11. S11. S11.	-3,70 2,70 2,70 1.70	; JT#12/242
1M.OP	: LOOP BACK TO THE INNER LOOP		1004 111 1808 1819	2,R0 R0, EJT 2,R0 1.R0	; UT#2/2+2
			115	80,€JT 2,80 1.80	; JT#2/2+2
			1878	2,80	
			ij 2	1,18	T) carron
SPECIAL BUTTERFLY FUR MEU				2	
			<u>5</u>		י ולאוביאו :
LDI IRI.ARA			£	8	THE STATE OF THE S
	· POINT TO SIN(45)	•			
	COCATE COSTME TAMEN 484=CD21	- 88	E BIT-RE	DO THE BIT-REVERSING OF THE OUTPUT	5
	י מודעור מסוער זומרע עני ספי	•			
		. 6	2	ACCTOTA DE	11 6
		į	3 8	J. 1311.	. OF CHAIR DE ONE 1500 THAN DESTREET B
ADDF +4R2, *AR0, R1	; R1=X(I)+X(I2)		ğ :	1,16	TO COLUMN OF THE PARTY OF THE P
SUBF +AR2, +AR0, R2	; R2=X(I)-X(I2)		Ē	DH1, 11C1 14D	# 11 5 371C-ONT :
	. R2=V(1)+Y(12)		Ē	EINPUT, ARO	
200 000 000 000	D4-V11-V12)		Ē	PINPUT, ARI	
	(71) - (1) - (1)	•			
	; KO=K(11)+K(13)		5	orto	
SUBF RI, PS, PS	14 CF 22 ··		2	D1 124	
ADDF R5.R1	. R1=11+5		<u>=</u>	ARO, ARI	
	. DSev(11)+v(13)		뿚	DOM:	
			<u></u>	#ARO. RO	
	25		1 2	10 10	
	: K3#C3#C	=	5 8	14.1	
STF R3, #+480	; Y(1)=K3+K5		<u>+</u>	KO, ##KI	
STF R1 =480++(1R0)	: X(1)=01+02	::	SIF	R1, #80	
	. 01=Y(11)-Y(13)		'n	#+4R0(1),R0	
	. 02/4/11/24/12	::	Š	#+#R1(1),R1	
	1 NOTICE 11.00		STF	RO. 4+48(1(1)	
	14 CH (11) 1	:	21.0	P1 4+490(1)	
STF R7, #4R1 ++ (1R0)	5 (11) x :	- 1	<u>.</u>	10,000	
ADDF R3.R2.R5	: R5=R2+R3	884	è	(7)030++4	
	. K2+-12+63	BITRV	Š	##G1 # (180) B	
Came pr 94 92	19-70-64	•			
	1070-90	* REST	36.36	RESTORE THE REGISTER WALUES AND RETURN	RETURN
ADDF RI,R4	: K4=K4+K1		!		

:: ::

:: :: :: :: II •

£ £ £ £ 5 5 5 5 5 F

Appendix C.Radix-2 Real FFT

Appendix C1. Generic Program to Do a Radix-2 Real FFT Computation on the TMS320C30

THE PROGRAM IS TAKEN FROM THE PRESENCE IN ET., JAME 1997 ISSUE THE PROGRAM IS TAKEN FROM THE PRPER BY SORDERSH ET AL., JAME 1997 ISSUE THE PROGRAM IS TAKEN FROM THE PRPER BY SORDERSH ET AL., JAME 1997 ISSUE COUNTY THE ORGAL DATA RESPONDET HE INTERNAL PERFORM. THE COPPUTATION IS DONE IN-ALLIED IN A SCHOOL DIRECTIVE AND SPECIFIED DATA RECEDENCE WATER OF THE PROGRAM. THE THALLIED IN A SCHOOL DIRECTIVE AND SPECIFIED DATA IS THE POLITION. THE DATA IS NA. A NA. A = NA.2. ANTHRES PARAMISE FILE TO PRESENTE THE THE AND LOCKION AND THE PRESENTE HE CHECKION AND THE THALLIED IN A SCHOOL DIRECTIVE AND SPECIFIED DATA IS NOT A NA. A = NA.2. ANTHRES PARAMISE FILE TO PRESENTE THE THE MOLITION OF THE PROGRAM IS SCHOOL OF	OPPI APRI, APO ; YOHWIGE LOCATIONS ONLY BGE CONT ; IF AROCARI I'M ADD BA				LENGTH-TNO BUTTERFLIES	the Tribute of the Tr	IRO,RC	SUBI 1, RC ; RC SHOULD BE ONE LESS THAN DESIRED &	BUKI	AUG ++ARO, +ARO++, RO ; RO#X(1)+X(1+1)	SIF RO ARO	STF R1, #AR0++		FIRST PASS OF THE DO-20 LOOP (STAGE K=2 IN DO-10 LOOP)	LDI @INPUT,ARO : ARO POINTS TO X(1)		LSF -2.RC REPEAT N/4 TIMES	-	BLK2	FALCE 6+6470 (1RO), 64270+(1RO), RO ; RO=X(1)+X(1+2) SIBE 6450 6-650 (1RO) R1 , R1=X(1)-X(1+2)		STF RO, +-ARO(IRO) ; X(I)=X(I)+X(I+2) STF RI #ABO++(IBO) ; X(I+2)=X(I)-X(I+2)	STF RO, #+ARO		MAIN LOOP (FFT STAGES)	LDI BEFTSIZ. IRO	-2, IR0	••	LSF -1, IRO	••	LSH 1, R3 ; N2=24N2	INNER LOOP (DO-20 LOOP IN THE PROGRAM)	9	
THE SAME FROM THE PAPER BY SOREHEEN ET AL., JAME 1997 ISSUE CUTIONS ON ASSU. FILT REVERSAL IS DONE AT THE RECONNITION ON THE TREADON. FILT REVERSAL IS DONE AT THE RECONNITION TO THE PROGNAM. ACTORS ARE SUPPLIED IN A TRALE BY IT IN AND LOCKIN, ARE ACTORS ARE SUPPLIED IN A TRALE BY IT IN AND LOCKIN, ARE FLORE SAME PROPOSE, THE SIZE OF THE FIT IN AND LOCKIN, ARE IN A TAN A = N/2. F. PAPAMILIDAL IS FRT SIZE IN STRUKHIS IN ST		**		STIE .	• •	* 9.	. "					==	•	• •	•			•	•					•	• •	•			d007			• •		14.06
WHY TO DO A RADIT IS TAKEN FROM THE CITUTIONS ON ASSET. IN THE RESIDE IN THIM THE SAME PROPOSE, ACADOR ONE CITUTION ON A SEPRENCING MAY A HAVA = NAZ. ACADOR ONE CITUTION ON A SEPRENCING MAY A HAVA = NAZ. ACADOR ONE CITUTION ON A SEPRENCING MAY A HAVA = NAZ. INC. PAPPANICHALIS INC.		2 REAL FFT COMPUTATION ON THE THS320C30	PAPER BY SORENSEN ET AL., JUNE 1987 ISSUE	Second or sectional and vicinity in second second	DONE AT THE BEGINNING OF THE PROGRAM,	ED IN A TABLE PUT IN A . DATA SECTION, THI	E FILE TO PRESERVE THE OENERIC NATURE OF T	, THE SIZE OF THE FFT IN AND LOG2(N) ARE AND SPECIFIED DURING LINKING, THE LENGTH (; ENTRY POINT FOR EXECUTION	: FFI 51.02	: ADDRESS OF SINE TABLE	. HENDRY WITH THEM !	: MEMORY WITH OUTPUT DATA				; STARTING LOCATION OF THE PROGRAM	; RESERVE 100 MORDS FOR VECTORS, ETC.						; COMPAND TO LOAD DATA PAGE POINTER	GINNING	11 S	RC SHOULD BE ONE LESS THAN DESTRED	: IROHALF THE SIZE OF FFT=N/2		
		,	w	5	S	تج	\$	¥ ₹	N/2.	HAL IS	S						₹												3	٤		<u>8</u>	ê ş	į

```
BRANCH TO ITSELF AT THE END
                     8
            . 3
                                                                                                                                                                                                                             ; IR1=SEPARATION BETWEEN SIN/COS TBLS
                                                     ; AR2 POINTS TO X(12)=X(1-J+M2)
; AR4 POINTS TO X(14)=X(1-J+N1)
                              ; AR3 POINTS TO X(I3)=X(I+J+N2)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              : LOOP BACK TO THE INNER LOOP
                                                                                                                                                                                                                                                                                                                                             : R0=-X(I3)+SIN+X(I4)+00S
        ; ARI POINTS TO X(II)=X(I+J)
                                                                                                                                                                                                                                                                                                                        : R2=X(13)+COS+X(14)+SIN
                                                                                                                                                                      : X(I+N4+N2)=-X(I+N4+N2)
                                                                                                                                                           ; X(I+N2)=X(I)-X(I+N2)
                                                                                                                                                                                                                                                                                                                                                                                x(13)=-x(12)+R0 !!!
                                                                                                                                                                                                                                                                                                                                                                                                       x(14)=x(12)+R0 !!!
                                                                                                                                                                                                                                                   REPEAT IM-1 TIMES
                                                                                                                         ; X(I)=X(I)+X(I+N2)
                                                                                                                                                                                                                                                                                                                                                         : R1=-X(12)+R0 !!!
                                                                                                             : RO=-X(I)+X(I+N2)
                                                                                                                                                                                                                                                                                                                                                                    R1=X(12)+R0 !!
                                                                                                   ; R1=X(1)+X(1+M2)
                                                                                                                                     : R0=X(1)-X(I+R2)
                                                                                                                                                                                                                                                                                                                                                                                                                            x(11)=x(11)+R2
                                                                                                                                                                                                                                                                                                                                                                                                                                        x(12)=x(11)-R2
                                                                                                                                               : R1=-X(1+N4+N2)
                                                                                                                                                                                                                                                                                                                                   HR3, #AR0++(IR0), R0 ; R0=X(I3) *SIN
                                                                                                                                                                                                                                                                                                 #AR4, #AR0,R1 ; R1=X(14) #SIN #AR4, #+AR0(1R1),R1 ; R1=X(14) #COS
                                                                                                                                                                                                                                                                                      HR3, #+AR0(1R1), R0 ; R0=X(13) #COS
                                                                                                                                                                                                                                                                                                                                                                                            R1=X(11)#R2
                                                                                                                                                                                                                                                                                                                                                                                                                  R1=X(11)-R2
                                                                                                                                                                                                                                                                                                                                                                                                                                                                          : ARS=I±1
                                                                                       : R0=X(I)
                                                                                                              RO, #++AR5(IR1), RO
                                                                                                   *** PES ( IR1 ) , RO, R1
                                                                                         MARS++(IR1), R0
                                                                                                                                               R1, *-485(IR1)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         1,R5
@LOGFT,R5
LOOP
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      FFTS12, ARS
                                                                                                                                                                                                                    BFFTS17, IR1
                                                                                                                                                                                                                                                                                                                                                           ##R2, R0, R1
                                                                                                                                                                                                                                                                                                                                                                     HARZ, RO, R1
                                                                                                                                                                                                                                                                                                                                                                                            *AR1,R2,R1
                                                                                                                                                                                                                                                                                                                                                                                                                  R2, ##R1, R1
                                                                                                                                                                                                                                                                                                                                                                                                                                                               EINPUT, ARS
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           EINPUT, ARS
                                                                   R3. AR2, AR4
                                                                                                                                                                                                                                                                                                                                                                                 K1. ##83++
                                                                                                                                                                                                                                                                                                                                                                                                                             R1, ##R1++
                                                                                                                                                                                                                                                                                                                                                                                                                                         RI. ##R2-
                                                                                                                                                                                                                                                                                                                                                                                                        R1, +AR4-
                                                                                                                                                                                                                                                                                                                         RO, R1, R2
                                                                                                                                                                                                                                                                                                                                                RO, R1, R0
                                                                                                                                                            RO, ##R5
                                                                                                                                                                                                                                                                                                                                                                                                                                                                          2. PS
                     AR1, AR3
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                9
                                            M3, AR2
                                                                                                                                                                        R1, ##85
                                                                                                                                                                                                                              -2, IR1
                                                      2,442
                                                                                                                                                                                                                                          7.
E.R.
           ž
                                                                                                                                                                                                                                                     2,80
                                                                                                                                                                                             INNERNOST LOOP
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                5
                                                                                                                                                                                                                                                                                                                                                           99 86 FS
                                                                                                                                                                                                                                                                                                                                                                                                       18 E
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      Ĕ
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          # & &
                                                                                                                                                                                                                                                                                                                        ğ
三季三日
```

.. ¥ ..

Appendix C2. fft_rl-Radix-2 Real FFT to Be Called as a C Function

	•		
	· er •	P	
: ffl_r Radix-2 recal FT 10 BE CALLED AS A C FUNCTION.	.0.080 .0.081	JFT.A.	; ENTRY POINT FOR EXECUTION ; AUDRESS OF SINE TABLE
SIS: int fft_i(M, M, data) int M PTET SIZE NEWS int M WHEEN OF STACES = LOGZ(N) float *data ABBON WITH INFUT AND OUTPUT DATA		FTS12, 1 LOGFT, 1 INPUT, 1	
IPTION: GENERIC FUNCTION TO DO A RADIL-2 FFT COPPUTATION ON THE TRS20C2O. THE DATA ARRAY IS H-CLOGG, WITH OMY FACE, DATA, THE CUTPUT IS STORED IN THE SAME LOCATIONS WITH REAL AND INICIDENT POINTS R AND 1 KS FOLLORS: RIO, RITI,, RINZ.), ILVZ-1),, IN THE PROPER BY SURBIGISH FOR PROCESSION THE FORTIAM PROCRAM THE PAPER BY SURBIGISH FOR THE WEST ISSUE OF TRANS. ON ASS., THE COMPUTATION IS DONE IN-CLAKE, AND THE ORIGINAL DATA IS DESTROYED. BIT RECURSAR IS IN-CLERKIED AT THE REGINAL DATA IS DESTROYED. BIT RECURSAR IS IN-CLERKIED AT THE REGINAL DATA IS DESTROYED. BIT RECURSAR IS	SINTRBword .SINK INITIALISE C.FUNCTION	SINE G G G G G G G G G G G G G G G G G G G	; SAME DEDICATED PEGISTERS
THE SIME/COSINE TABLE FOR THE THILDER FACTORS IS EPECTED TO BE SUPPLIED DORING LINK TIME, AND IT SHOLLD HAVE THE FOLLOWING FORMAT: - globalsinedatasinefloat valuel = sin(0e26p1/N)float value2 = sin(1e26p1/N)		LD ++P(2), R0 1 1 1 1 1 1 1 1 1	; NOE MOURAIS IN THE PROGRAM ; THE NAMES IN THE PROGRAM COMMITTEE
.float value(N/2) = cos(NV4)+24pi/N) THE VALUES value! TO value(N/4) ARE THE FIRST QUANTER OF THE SINE PERSON AND value(N/41) TO value(N/2) ARE THE FIRST QUANTER OF THE		6FFTS17,RC 1,RC 6FFTS17,IRO -1,IRO	; RC-N ; RC SHOULD BE ONE LESS THAN DESIRED # ; IRO-HAUF THE SIZE OF FFT-M/2
USING STRUCTURE LPON THE CALL: FP(4) MT FF FP(3) N TE		EINPUT, ARO EINPUT, ARI BITRV ARI, ARO CONT	; KOMMOE LOCATIONS DALY ; IF ANOCARI
-FP(1) : RETURN ALOR : -FP(0) : OLD FP : -FE(1STERS USED: RO, R1, R2, R3, R4, R5, ARO, AR1, AR2, AR4, AR5, IRO, IRI, R5, R6, RC AUTHOR: PANOS E. PROPRIEDLE.	1	##RO, RO ##RI, RI RO, ##RI RI, ##RO ##RO++ ##RI++(IRO)B	
TEJAS INSTRUMENTS 00708EN 13, 1997	161 161 1888	EINPUT, ARO IRO, RC 1, RC	; ARO POINTS TO X(1) ; REPEAT N/2 TIMES ; RC SHOULD BE ONE LESS THAN DESINED #

; R1=-X([+M4+M2) ; X([+M2]=X([)-X([+M2)	; X(I+M4-N2)=-X(I+M4+N2)			S MIT STOWNED STATES STATES .		; REPEAT N4-1 TIMES		. po=/(12)*COS			; R2=X(13)+COS+X(14)+SIN	HAR3, #ARO++(IRO), RO ; RO=X(I3)+SIN	; RO=-X(13)+SIN+X(14)+COS !!!	; R1=-X(12)+R0 !!!	; R1=X(12)+R0 !!!	; X(13)=-X(12)+R0 :::	: KI-AIII/462 . VIII)-WIII/462	· RI=X(11)_H2	; X(II)=X(II)+R2	; X(12)=X(11)+R2			; ARS=1+N1		; LOOP BACK TO THE INNER LOOP							S AND RETURN										
**************************************	R1, +485	٠		#FFTS12, IR1 -2, IR1	R4,RC	2,RC	2	#407 #±400/1011 DO	#484 #480-R1	#4R4, #+4R0(1R1), R1	RO, R1, R2	+AR3, +AR0++(1R0)	RO, R1, R0	+4R2, R0, R1	**#C2, R0, R1	K1, #98(3**	PHR1, R2, R1	10 mg	RI, +4RI ++	R1.##R2		EINPUT, ARS	R4, AR5	CMY, ZIST 140	180	ellerut, Arco			£,	er net		RESTORE THE REGISTER WALLES AND RETURN	ş	2 4	102	*	£					
STEGE	STF	INNERNOST LOOP		<u>5</u> 5	5	SUBI	e e	2	ž	H.	ADDF	HPYF.	SUBF	ABS.		4 P	į	3				3081	4001	1	BLTD iii	<u> </u>	2 2		100	5	2	RESTONE	8	2 8	2	æ	æ	R ETS				
==		* *	•				•				==				:	::	:	=	==	BLK3	•							•			•	•	•									
	; R0=X(I)+X(I+1) . R1=X(I)-X(I+1)	; X(I)=X(I)+X(I+1)	* X(I+1)=X(I)-X(I+1)	E K=2 IN DO-10 LOOP)		; ARO POINTS 10 X(1)	ZH-Z-NIT :	: REPEAT N/4 TIMES	RC SHOULD BE ONE LESS THAN DESIRED #			RO), RO ; RO=X(1)+X(1+2)	; R1=X(I)-X(I+2)	; RO=-X(1+3) ; V(1)-=V(1)+V(1+3)	. x(1+2)=x(1)-x(1+2)	: X(1+3)=-X(1+3)					; IRO=INDEX FOR E	SALES THE CURRENT STATE NUMBER		. 1=1.	- M-2-4-4	: N2=2*N2	. :		· ARS POINTS TO X(I)		; ARO POINTS TO SIN/COS TABLE	; IR1##		; AR1 POINTS TO X(II)=X(I+J)	CALL AT VA-101 V OT STATOS COA	HAS FULLIS IN ALLS/-ALLYSTICA	. AR2 POINTS TO X(12)=X(1-JHM2)	+ APA POINTS TO X(14)=X(I-J+N1)	1137	; NJ=X(1)+X(1+NZ)	: R0=-X(1)+X(1+N2)	: X(I)=X(I)+X(I+X)
BUCI	#+ARO, #ARO++, RO	FO, = -480	R1, #480++	FIRST PASS OF THE DO-20 LOOP (STAGE K=2 IN DO-10 LOOP)		EINPUT, ARO	6FFTS17.RC	-2.RC	1,80		BUC	*+ARO(1RO), *ARO++(1RO), RO	#ARO, #-ARO(IRO), R1 ; R1=X(I)-X(I+2)	PA 4 ABO(100)	R1 +480++(180)	RO *+4RO	•	T STAGES)		GFFTSIZ, IRO	-2, IR0	Ç.	 2	9	1 R4	5	-	INNER LOOP (DO-20 LOOP IN THE PROGRAM)	PINPIT ARS	IRO, ARO	esintab, aro	R4, IR1	ARS, ARI	1,481	AR1, AR3	ACT ACT	2 Mez.	R3, AR2, AR4	A 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	M-665(181), R0, R1	RO, ***AR5(IR1), RO	R1. +-ACS(1R1)
er e	ADOF	# HS	SH.	TIRST PASS OF		5 :	3 3	3	1905		er i	A P		1	÷ #	15		MAIN LOOP (FFT STAGES)	:	ë	5 2	3	3 5			3			ä	ē	_	5	Ē	ADDI	9	3 2	i	Ş	į		38	STF
•		EK:	:: ,											=	2 :	::								٤	3					N.			•									::

Appendix C3. Generic Program to Do a Radix-2 Real Inverse FFT Computation on the TMS320C30

EINPUT, ACS 1, ACS POINTS TO X(1)	980	R4, IR1 : IR1 #4		1,ARI ; ARI POINTS TO X(II)=X(I4J)	PR ARCH COLUMN C	•	2.AR2 : AR2 POINTS TO X (12)=X (1-, HMZ)	. AR4		***ARS(IR1) ; POINT TO X(I+M4)	+-MS(IRI), +-MS(IRI), RO	FYMCOLDLI, F-MCCLDLI, RI RO #-ARS(IRI) . Y(I)=X(I)+X(I+MC)	_	**4K5,R0		RO, 4-AR5(IR1) ; X(I+M4)=24X(I+M4)	IX'(IXI)'XI	C-1.0.KI D1 #ADC141701 . (2014-2017) - (2014-2017)		4001		, IRI	-2, IRI ; IRI=SEPARATION BETWEEN SIN/COS TBLS	2.RC : REPEAT NA-1 TIMES			**************************************	R1, ++4R0(IR1), R0 ; R0=T1+CDS	••	+4R3, +4R4, R2 ; R2=T2=X(13)+X(14)	. 9	NZ, ************************************		R2, ++4R0(IR1), R6 : R6=T2+C0S		R1, +ARO++(IRO), RO ; RO=T1+SIN		RO, +4R4 ; X(I4)=T1+SIN+T2+COS	EINPUT, ARS	BFTS11, ARS	INLOP ; LOOP BACK TO THE INNER LOOP	einput, Ars	INU, HAYO ESINTAB, ARO ; ARO POINTS TO SIN/COS TABLE
9 5	Ş	9	5	9	3 §	Ē	88	Ā		ğ	S	, t	STF	Ė	i de	HS E	9 8	į	Š	INNERNOST LOOP		Ē	<u> </u>	i i i i i i		E		#PYF	STF	ADD	8	į į	*	#PYF	STF	HPYF.	8	STF	SUBI	8	H	Q	9
1007		INCOP	•						•					::		=	Ξ		•	*	•				•				==			=	=		==			.	•				
		GENERIC PROGRAM TO DO A RADIX-2 REAL INVERSE FFT COMPUTATION ON THE		THE (MEAL) DATA RESIDE IN INTERNAL MENDRY, THE COMPUTATION IS DONE	INTTURE, INC. BIT REVENSAL IS DURE AT THE BEGINNING OF THE PROGRAM, THE INPUT DATA ARE STORED IN THE ENLIGHTNE CORES.		N/2-1) IN(1)		THE THIDDLE FACTORS ARE SUPPLIED IN A TABLE PUT IN A .DATA SECTION. THIS	DATA IS INCLUDED IN A SEPARATE FILE TO PRESERVE THE GENERIC NATURE OF THE	PROGRAM. FOR THE SAME PURPOSE, THE SIZE OF THE FFT M AND LOGZ(M) AND	DETINED IN M. COURT DIRECTIVE AND SPECIFIED DURING LINKING. THE LENGTH UP. THE TABLE IS N/4 + N/4 = N/2.		DECEMBER 21, 1988		worthware and trained with a	ENINT FUINI FUN EXECUTION	3775 111 :	; ADDRESS OF SINE TABLE		; NEHORY WITH INPUT DATA					; STARTING LOCATION OF THE PROGRAM	; RESERVE 100 WORDS FOR VECTORS, ETC.						; COMMAND TO LOAD DATA PACE POINTER				: INCEINUEX FOR E	; NO HULLS THE LUMBER! STRUE MUNIBER	; R3=N1/2=N2		: R4=N1/4=N4		
2	8	C PROGRAM TO DO A RADIX-		EAL) DATA RESIDE IN INTE	INTICACE, THE BIT NEWENSHE IS JUNE AT THE BEG.		RE(0), RE(1),, RE(N/2), IN(N/2-1),, IN(1)		IDOLE FACTORS ARE SUPPLII	S INCLUDED IN A SEPARATE	1. FOR THE SAME PURPOSE,	THE TABLE IS N/4 + N/4 = N/2.		AUTHOR: PANCS PRPANICHALIS	TEXAS INSTRUMENTS	1331			CLOBL SINE		.BSS INP, 1024			17.6		.MONGO IFFT	SPACE 100		N :	MOKU II			P FFTS12		MAIN LOOP (FFT STAGES)		04.1.			I @FFTS17,R4		au	i
• APPENDIX CA		* GENERIC PR		# 15 S	T I I I I		. RE(0).	•	18 18	* DATA !:	# 190g	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		+ AUTHOR:	•	•	: •	ب ب	. •		~:		•	INITIALIZE		•	s.			a deliver			IFFT: LDP	_	DI NIM	•	3 5	3 5	3	9	s	* INMERLOOP	

; IF AROCARI	; BRANCH TO LITSELF AT THE BAD	
COMT #4870, RO #4871, R1 R0, #4811 R1, #4870 #4870++ (1R0)B	큡	
	3	
:: :: CONT	• a	
; E=f+2 ; M=M/2 ; M2=M2/2	HO-2-RC K(1-2) = K(1-2) HO-2-CONT(1+1) K(1-2-CONT(1+2) HO-2-CONT(1+2) HO-2-CONT(1+2) HO-2-CONT(1+2) HO-2-CONT(1+2) HO-2-CONT(1-2) HO-2-CON	
1, R5 6L00FF1, R5 L00P 1, 180 -1, R4 -1, R3	LIMET PRESS OF THE WALTH LOOPE LIDIT 2, 1800 LIDIT 2, 1800 LISH 2, 1800 LISH 3, 2, 1800 LIDIT 4, 44800 LIRO), RO PRTP 8 LIDIT 4, 44800 LIRO), RO LIDIT 5, 1800 LIDIT 5, 4800++ LIRO), RI 1, 4800++ RO LIDIT 6, 4800++ R	
100 8 8 3 3 4 5 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1	1.0617 PASS OF THE MAIN IL. 1.100 2.100 1.100	

Appendix D. Discrete Hartley Transform

Appendix D1. Generic Program to Do a Radix-2 Hartley Transform on the TMS320C30

18' 18' E		KI, #880		9 (DMT) ++ (TMD)	STI DEBLE		ATHERT AND AND TOTAL OF THE PARTY OF THE PAR		••	I, M. T. SAULD BE URE LESS 1998 DESIRED 8			######################################	••			FIRST PASS OF THE DO-30 LOOP (STAGE K=2 IN DO-20 LOOP)	A LY OT STRING AND A CAMP TICKLE		. BC	•	1, RC 1, RC SHOULD BE ONE LESS THAN DESTRED 8	2	#+480(180) #480++(180) 80 . BOEK(.)+X(12)	+ARO. +-ARO(IRO), R1 , R1=X(J)-X(L2)	RO, +- 4RO(IRO) ; X(J)=X(J)+X(L2)	•	NO.4-MNO.KI ; RI=X(L3)+X(L4)	 20		•		T STAGES)	#FHTSIZ.IRO	•	••	1,84	•		1,R4 ; M4=2+M4	1,R3 ; N2=24N2		INNER LOOP (DO-30 LOOP IN THE PROGRAM)	EINPUT, ARS 1: ARS POINTS TO X(J)		980	R4, IRI : IRI #4
5	#5	<u>,</u>	1	È	SOUCHALTHO BUTTERS 150		ä	j 5	į	į				į	. HS	;	ST PASS OF	=	3 5	i	5	SUBI	8	4	SUBL	STF	<u>.</u>		#	SF	SIF		MAIN LOOP (FAT STAGES)	ĕ	西	ë	ë	ë	<u>5</u>	西	西		100 a	ë	ë	100	5
==	Ξ	: 1		DI 180	• •	•	•			•	•			2	i ::		*	•					•				::	=	-	==	BLC2	•	₹ • ·	•					6 007			+	<u>.</u>	•	INCOP		
		UEMEKTIC PNUJARAN 10 DO A RADIT-2 HARTLEY TRANSFORM ON THE THS320C30.		THE PHOUMEN IS THREEN FROM THE PRIPER BY SCHEDISEN ET AL., OCT 1965 ISSUE			THE VICINITY OF BIT STATEMENT OF THE CONTROL OF THE	LUNE HI INE BEGINNING UP THE PROLUMEN.		MAY IS THE INC. OF THE SUPPLIED IN A LABOR THE	FILE TO PRESENTE THE GENERALC MATURE OF THE	THOUSEN, THE SHIP THOUGH, THE SIZE OF THE FAIL MAND LOGICAL AND	DETINED IN HIGHDOOD DIRECTIVE AND SPECIFIED DURING LINKING, THE LENGTH OF THE TABLE TO MAY A MAY A MAY.		OBS. TI COMEDIA	11, 1700		; ENTRY POINT FOR EXECUTION	Wiczel .	: ADDRESS OF SINE TABLE		; MENORY WITH INPUT DATA					; STARTING LOCATION OF THE PROGRAM	THE SECTION WITH STATE OF THE SECTION OF THE SECTIO						; COMPAND TO LOAD DATA PAGE POINTER			:	7	: NC SHOULD BE ONE LESS THAN DESIRED .		: INCHMUL THE SIZE OF FHI=N/2				; XCHONGE LOCATIONS ONLY	; IF AROCARI	
ä		MUJORNA 10 DU A RADIX-2		OFFI IS HAKEN FACED THE P	UF THE THORISACTIONS ON ASSP.	PATA OFFICE IN THE	THE BIT OFFICE AND TO BE	INC. DI PENEROME, 13 D	TO STAND AND STANDS IN	ALE PRESUND ME SUPPLIE	INCLUDED IN A SCHOOL	TUR THE SHIP FUNDING.	DEFINED IN HOUSE DIRECTIVE R		AUTHOR: PRINGS PRPRINCHALIS	TEXAS INSTRUMENTS				SINE SINE		1MP, 1024	!		ш		E .	100		*			1	FHTS12		LU INE BIT NEVERSING AT THE BEGINNING		CFHISIZ, RC	1,40	G-HIS11, 1KO	-1, 180	ATMENT AND	EINE OI, PRII			CONT	24,5
APPENDIX DI	į,	FIERE	9		# #	9			91111	7 VI VIV	21 200				UTHOR: P	_	į	1 0		G.08		S.	TEXT		INITIALIZE			SPACE.		FHTS12 . MORD		SINING MORE	į	ŝ		±		i i	9	3	5 5	j	i	9	8	썵	j

```
BRANCH TO ITSELF AT THE DID
ĝ
            . 8
                                                                                                                                                                                                                                                                           ; IR1=SEPARATION BETWEEN SIN/COS TBLS
                                                                ; AR2 POINTS TO X(L2)=X(J-1+1+N2)
; AR4 POINTS TO X(L4)=X(L2+N2)
                     ; ARI POINTS TO X(L1)=X(J+I-1)
                                           ; AR3 POINTS TO X(L3)=X(L1+N2)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            : LOOP BACK TO THE INNER LOOP
                                                                                                                                                                                                                                                                                                                                                                   ; R2=X(L3)+C0S+X(L4)+SIN=T1
                                                                                                                                                                                                                                                                                                                                                                                          ; R0=X(L3)+SIN-X(I4)+C0S=T2
                                                                                                                                                                                                      ; X(L3)=X(L3)+X(L4)
                                                                                                                                                                                                                  ; X(L4)=X(L3)-X(L4)
                                                                                                                                                                                                                                                                                                REPEAT IM-1 TIMES
                                                                                                                                                          ; X(L2)=X(J)-X(L2)
                                                                                                                                    ; X(J)=X(J)+X(L2)
                                                                                                                                                                                 ; R1=X(L3)+X(L4)
                                                                                                                         ; RO=-X(J)+X(L2)
                                                                                                                                                                                           ; R1=X(L3)-X(L4)
                                                                                                                                                                                                                                                                                                                                                                                                                            ; X(L4)=X(L2)-T2
                                                                                                                                                                                                                                                                                                                                                                                                                                                  : X(L2)=X(L2)+T2
                                                                                                                                                                                                                                                                                                                                                                                                                                                                       ; X(L1)=X(L1)+11
                                                                                                              : R1=X(J)+X(L2)
                                                                                                                                               R0=X(J)-X(L2)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    : X(L3)=X(L1)-T1
                                                                                                                                                                                                                                                                                                                                                                               #AR3, #ARO++(IRO), RO ; RO=X(L3) +SIN
                                                                                                                                                                                                                                                                                                                                             ; RI=X(L4)*SIN
                                                                                                                                                                                                                                                                                                                                                         HRR4, #+ARO(1R1), R1 ; R1=X(L4) #CDS
                                                                                                                                                                                                                                                                                                                                  #AR3, #+AR0(IR1), R0 ; R0=X(L3) #COS
                                                                                                                                                                                                                                                                                                                                                                                                      ; R1=X(L2)-T2
                                                                                                                                                                                                                                                                                                                                                                                                                 R1=X(L2)+T2
                                                                                                                                                                                                                                                                                                                                                                                                                                                               R1=X(L1)-T1
                                                                                                                                                                                                                                                                                                                                                                                                                                        R1=X(L1)+T1
                                                                                                                                                                     ; RO=X(L4)
                                                                                                    : R0=X(J)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     : ABS=1±13
                                                                                                                         RO, *** ARS ( IR1 ), RO
                                                                                                              *+AR5(1R1), R0, R1
                                                                                                                                                                                 RO, #-ARS(IR1), R1
                                                                                                                                                                                            RO, +-AR5(IR1), R1
                                                                                                 #AR5++(IR1),R0
                                                                                                                                                                     ++AR5(IR1),R0
                                                                                                                                    R1, +-AR5(IR1)
                                                                                                                                                                                                        R1, *-AR5(1R1)
                                                                                                                                                                                                                  R1, ++4R5(IR1)
                                                                                                                                                                                                                                                                                                                                              HAR4, +ARO, R1
                                                                                                                                                                                                                                                               BFHTS12, IR1
                                                                                                                                                                                                                                                                                                                                                                                                                                     ##R1, R2, R1
R1, ##R2--
                                                                                                                                                                                                                                                                                                                                                                                                                                                            R2, ##81, R1
R1, ##81 **
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                PFHTS12, ARS
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  1, R5
BLOGFHT, R5
                                                                             R3, AR2, AR4
                                                                                                                                                                                                                                                                                                                                                                                                      RO, #MR2, R1
                                                                                                                                                                                                                                                                                                                                                                                                                 #M2,R0,R1
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        LINPUT, ARS
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      INPUT, ARS
                                                                                                                                                                                                                                                                                                                                                                                                                           R1, +AR4--
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  R1, #483++
                                                                                                                                                                                                                                                                                                                                                                     PO, R1, R2
                                                                                                                                                                                                                                                                                                                                                                                          R1, R0, R0
                               AR1, AR3
R3, AR3
                                                                                                                                                          £4.0£
                                                      AR3, AR2
           A65, A81
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     R3.485
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           14.00
                                                                                                                                                                                                                                                                           -2, IR1
                                                                 . AR2
                      ã.
                                                                                                                                                                                                                                                                                     38,
                                                                                                                                                                                                                                                                                               2, RC
                                                                                                                                                                                                                                          INNERNOST LOOP
           ₹ēēē
                                                                                                                                                                                                                                                                                                                                                        A SEC
                                                                                                                                                                                                                                                                                                                                                                                         SUBF
SUBF
ADDF
STF
STF
STF
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               ĕ
                                                                                                                                                                                                                                                                                                                                                                                                                                                                         .. 2
```

::

::

::

Appendix E. Discrete Cosine Transform

Appendix E1. A Fast Cosine Transform

; TWO BUTTENCLIES ANE CALCULATED AT ; THE SAME TIME.	; OET LOMER HALF OF EACH BUTTEMELY.	COMPANIES LATER)	SUBTRACT SECOND BUTTERFLY DATA.	; SUBTRACT FIRST BUTTEDFLY DATA.	; MULTIPLY 2ND SUBTRACTION RESULT BY	; COSTINE COEFFICIENT. AND SECOND	BUTTERLY DATA.	THE THE SABINACTION PESOL I	COUSINE CLEFFICIENT, ALLO FINST		••	AMOUNTAIN IN HOOSE AND SECTION Y	i motition in order 240 bollowell.		; 2ND BUTTEPFLY, SAVE 1ST ADDITION	: IN UPPER 1ST BUTTERFLY.		P SENIES.	; UPDATE REPEAT COUNTER FOR NEXT BLOCK	••	•-	HAVE BUTTERFLIES BEEN COMPLETED?	; DELATED DOWNA, IF NOT.	- •-	: UPDATE COSINE COEFFICIENT POINTER.	SET REPEAT MODE. (FASTER THAN USING	RPTB MAEN START AND END ADDRESS	; AME STILL (5000)		E.100P.		; UPDATE INDEX REGISTER. (DIVIDE BY 2)	; REINITIALIZE DATA POINTERS.		· IS FIRST BUTTERELY SERIES COMPLETED	DELAY BRANCH, IF NOT.	; MULTIPLY 2'S POWER COUNTER BY 2.		, POINTERS.	; SET REPEAT COUNTER FOR REPEAT BLOCK.		
•	24,2784	etters, K3	#AR3, #AR4, R1	#MR2, #MR1, R0	R1, *****R7, R1	Z, ##4, Z	90 400	M, 1 M, 10	KZ, ##KI, KZ	P. *AD TD	M1, ************************************	A LINE LANGE COL		RO, +AR3++(IR1)X	R2, #4R1++(IR1)X		01 2000 000 1	END OF CONTEX LUOP OF FIRST LUOP SENIES.	IRO, ARS, RC		##R3++, ##R2, R0	AR3, AR2	##R1++ ##R4 R0		2, AR7	0100H, ST				DELAY BRANCH FROM HERE TO NIDDLE_LOOP.		-1, IRI	A86, AR1	1R1, 487	2. IRI	OUTSIDE LOOP	1,465	IRO, AR4, AR3	;	IRO, ARS, RC	LOOP SERIES.	FINAL BUTTERFLY STAGE LODP.
OUTSIDE_LOOP: MIDDLE_LOOP: *	5 !	:	SUBF3	SIBES			•	CLI M		•	i i		END CENTER LOOP:	STF	:: SIF	•			A0013	•	ADDF3	- E	ADDES	•	ADDI	8	•		•	# DELAY BRANCH		कु	5		8	BCTD	₹	SUBIS	•	A0013	* END OF FIRST LOOP SERIES.	* FINAL BUTTERS
		ED BY BYEONG GI LEE IN HIS ANTICLE, FCT - A	ED IN THE PROCEEDINGS OF THE IEEE INTER-	ICS, SPEECH, AND SIGNAL PROCESSING, SAN	28A.3/1-4 VOL. 2, (CM1954-5/84/0000-0299).	The state of the s	TELL TO MELLUM MATURAL UNDER TIME LUMBING.	and district the common of the	MT AAS 1 STUT GOOD SOCIOUS THE WE TAKE STUT	NI ME CO COMPANIE CONTROL IN THE TANK AND				; FAST COSINE THANGSONN ENTRY POINT.	; LENGTH OF DATA BUTRY.	: TABLE OF COSINE COEFFICIENTS.	; IMBLE UF INPUT DATA.							: LOND DATA LENGTH.	SET BLOCK SIZE FOR CIRCLIAR	; ADDRESSING,	: LOAD DATA POINTER.	; LOND COSINE TABLE POINTER.	; INITIALIZE INDEX REGISTERS FOR FIRST	; BUTTERFLY SERIES.	; INITIALIZE DATA POINTERS.			. INITIALIZE 2/S POLED CHANTED	· FINISH DATA POINTER INITIALIZATION		; RC SHOULD BE ONE LESS THAN COUNT	; DESIRED.			BUTTERFLY STAGES EXCEPT THE FINAL ONE.	
	a fast cosine transform	BASED ON THE ALCORITHM OUTLINED BY BYEONG 61 LEE IN HIS ARTICLE, FCT - A	FAST COSINE TRANSFORM, PUBLISHED IN THE PROCEEDINGS OF THE IEEE INTER-	NATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, SAN	DIEUJ, CA, 19-21 MARCH 1984, P 28A.3/1-4 VOL. 2, (CH1954-5/84/0000-0299).	THE LAC DETAIL MODIFIED TO ALL SELECTION AND	CEES S HEOMETINE HAS BEEN FRONTEED TO MELLOW MATCHING COURT TIME LUMBAIN.	ייייי בייייי בייייי ביייייי ביייייייייי	THE ERECUENCY IDMAIN CORPETITIONS ARE IN DITIONAL RECUERS.	PLACE CALCULATION		AUTHOR: PAUL WILHELM		FCT ; FAST COSINE TRANSFORM ENTRY POINT.	••	 90	WELF : MALE UF INPUT DATA.				COS. TAB	WEFF				٠	C_DATA,AR6 ; LOND DATA POINTER.	٠.	••	••	••	AR6, AR0, AR2	1,462	I ARS INVITED OF PRICED COUNTY			••	••	•	FIRST LOOP SERIES	THIS LOOP SERIES DOES ALL THE BUTTERELY STAGES EXCEPT THE FINAL ONE,	FAN CONTRA I MP

MODI 1, APR		3.81 1, RC NOP + +840++(180)B ; COMITINE LD1	1, RC 1,
-1,485 -1,485 -1,485 -1,485 -1,485 -1,486 -1, 10,141,12 REPEAT COLANTER1,1487 -1,1488 -1,1487 -1,1488 -1,1487 -1,1488 -1,1487 -1,1488 -1,14		1, R. 6862+(1R0) B 9872, 4873 B 102, 1R5 IE 9872, 4873 B 102, 1R5 IE 9873+(1R1) Z, R 9873+(1R1) Z, R 9873+(1R1) Z 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	; CONTINE UPDATING POINTERS; THO AUDITIONS ARE DONE IN 1 1 400 FIRST THO DATA. ; SAME FIRST AUDITION. ; SAME SECOND SAMEOLY, IT NOT. ; SET IN SECON
100,485, RC		### 1992 #### 1992 #### 1992 #### 1992 #### 1992 #### 1992 #### 1992 #### 1992 #### 1992 ##### 1992 ##### 1992 ##### 1992 ###### 1992 ##########	; This Additions Net Done IN E.; This Additions Net Done IN E.; SAVE FIRST TAUDITION. ; SAVE SECOND ADDITION.
##07, +##07, RM CALCLAL CAN-0056(1744). ##02, -##1, RM CALLED SERTION) AT THIS EDOL. 300. 200. LOOP TOO BOTTOPELLES ARE CALCLAL + ROLD SERTION		BIO_INSIDE =#R1, =#R2+(IR1)%, RC =#R3, =#RR++(IR1)%, RC =#R3++(IR1)% R1, =#R3++(IR1)% R1, =#R3++(IR1)% R1, =#R3++(IR0)B, =#R2++(IR0)B, =#R2++(IR0)B, =#R2++(IR0)B, =#R2++(IR0)B, =#RR2++(IR0)B, =#RR2	ADD FIFE SECON THE SECON T
(ALLED S, SEGUR)	1977	### ##################################	; THO ANDITIONS ARE DONE IN EI 1 AND SECOND THO DATA, 5 SAVE FIRST ANDITION, 1 SAVE SECOND ANDITION, 1 GROUPS OF LIPORTE DATA POSITION, 1 (180.18, PO. 1, LPDATE DATA POSITION), 1 ST THIS LOOP COMPLETE?
1.00P		+481, +482++(1R1)1, R +483, +484++(1R1)1, R R0, +481++(1R1)2 R1, +483++(1R1)2 R1, +483++(1R1)2 +481++(1R0)8, +482++(+4818-484+484+484+484+484+484+484+484+484+48	AND FIFE SECONDS SECON
1.00P. 1.0P. 1.00P. 1.		+8R1, +8R2++(IR1), R +8R3, +8R4++(IR1), R R0, +8R1++(IR1), R R1, +8R3++(IR1), Z ELOOP FOR LAST LOOP SE +8R3++(IR0)B, +8R2++(+ +8R3++(IR0)B, +8R2++(+ +8R3++(IR0)B, +8R4++(+ +8R3++(IR0)B, +8R4++(+ +8R3++(IR0)B, +8R4++(+ +8R3++(IR0)B, +8R4++(+ +RR1++(IR0)B, +RR1++(+ +RR1++(IR0)B, +RR1	ADD FIRST FEED BY NO. 25CO
### ### ### ### ### ### ### ### ### ##		### ##################################	AND SECONDARY LAND SE
### 4#83,81 SUBTRACT 200 BUTTEREY DATA ### 180,744,70 ### 181, ### 181,70 ### 181, ### 181,70 ###		RI, #RE1++(IRI)7 RI, #RE3++(IRI)7 RI, #RE3++(IRI) B. #RE2++ #RE3++(IRO) B. #RE2++ #RE3++(IRO) B. #RE4++ #RE3++(IRO) B. #RE4++ #RE3++(IRO) B. #RE4++ KR1++(IRO) B. #RE4++ KR1++(IRO) B. #RE4++ KR1++(IRO) B. #RE4++	E 5800 9,88 9,88 1,88 1,80 1,80 1,80 1,80 1,80 1,80 1
### ### ### ### ### ### ### ### ### ##		RI, +483++(IRI)X FELOUP FOR LAST LOOP SS +482++(IRO)B, +482++(+482++(+682++(1RO)B), +484+++(+682++(1RO)B), +484+++(+682++(1RO)B), +482+++(+682++(1RO)B), +482+++(+682++(1RO)B), +482++(+682++(1RO)B), +482++(-482++(1RO)B), +482++(-482+(1RO)B), +482++(-482+(1RO)B), +482++(-482+(1RO)B), +482++(-482+(1RO)B), +482++(-482+(1RO)B), +482+(-482+(1RO)B), +482+(-482+(1RO)B	7E SECON
RI, AR, RI	7gg	RI, +483++(IRI)X FILODP FOR LAST LOOP SS +482++(IRO)B, +482++(+482++(1RO)B, +482++(1RO)B, +484+++ +4873++(IRO)B, +482+++ -4871++(IRO)B, +4872++(1RO)B, +4872+++	F SECO
RI, RH, RI *RRI,+HIRI), MACKA-HIRI), MACKA-HIRI), MACKA-HIRI), MACKA-HIRI), MACKA-HIRI), MACKA-HIRI), MACKA-HIRI), MACKA-HIRI), MACKA-HIRI) RR, +-MAC(IRI) R		R1, +883++1R1) X +881++1R0) B, +882++ +883++1R0) B, +884++ +8873++1R0) B, +884++ +881++1R0) B, +882++ R8, +881++1R0) B, +882++ R8, +881++1R0) B, +882++	FE SECON
### (+(R1), +462++(R1), R2 ; BY S. 400 IST B R3, +467, R3 ; MLITIPLY 200 40011104 RESUL R0, +462(R1) ; 7011, SAFE IST SEPRINCTION RESUL R2, +467, R2 ; MLITHY 1ST 40011104 RESUL R2, +461(R1) ; 7011 SAFE IST 8011764 R2, R3, R4, P3 ; 40011104 RLITHY 1ST 400111104 RLITHY 1ST 400111		ELOOP FOR LAST LOOP SE #REI+(IRO)B, #REZ+(#REI+(IRO)B, #REX+(#REI+(IRO)B, #REX+(#R. AREI LAST_INSIDE_LOOP	3,780 3,780 3,780 1,741
R3.+487, R3. R3.+487, R3. R3.+487, R2. R3.+487, R2. R4.11124 V3. SARE 1ST SABTRACTION RESULTING RESULT.	• • • • • • • • • • • • • • • • • • •	### (180) B, ### (3,78 3,78 3,78 3,78 1,78 1,78 1,78 1,78 1,78 1,78 1,78 1
RG, +-482(IRI) MULINY DAY BURDHOUS BOARD MULINY DAY BURDHOUS BOARD MULINY DAY BUTTERS IN A 4487 RZ MULINY DAY BURTERS IN A 4487 RZ MULINY DAY BARE DAY BURTERS IN A 4487 RZ MULINY DAY BARE DAY BURTERS IN A 4487 RZ MULINY DAY BARE DAY BURTERS IN A 4487 RZ MULINY DAY BUTTERS IN A 4487 RZ MULINY DAY DAY BURTERS IN THE PLY DAY DAY DAY DAY DAY DAY DAY DAY DAY DA	· R {	##R1+(IRO)B, ##R2++(#R9)B, ##R4++(IRO)B, ##R4++(IRO)B, ##R2++(IRO)B, ##R2++(IRO)B, ##R2++(IRO)B, #RA2++(IRO)B, #RA2++(IRO)B, #RA2++(IRO)B, #RA1PE_LOOP	CHRO1B.RO ; UPDATE DATA POINTERS. (1R01B.RO (1R01B.RO ; IS THIS LOOP COMPLETE? ; RELATED READOL; IF NOT.
RO, F-RECIER) 1. OUT. SHE SIS BUTTERN RESULT REVERSE AND ITIONS AT THE BIT REVERSE AND ITI	R {	#AR1++(IRO)B, #AR2++(#AR3++(IRO)B, #AR4++ #AR3++(IRO)B, #AR2+++ #AR1++(IRO)B, #AR2+++ R4, AR4 LAST_INSIDE_LOOP	
R2, +487, R2	9	##GF+(1RO)B, ##GF+ #AR3++(1RO)B, #AR2++(#AR1++(1RO)B, #AR2++(R4, AR4 LAST_INSIDE_LODP	(18018, RO (18018, RO (18018, RO 1 IS THIS LODE COPPLETE? 1 PELAYED BROWCH, IF NOT. 5 SET UP REPEAT COLATER.
RI, +RAF(RI) : .7071 SAVE 200 SUBTRACTION RALITIES. R2, +-RRILLIR1) : .200 SUBTRACTION RALITIES. R2, +-RRILLIR1) : SAVE 200 SUBTRACTION RALITIES. R3, +-RRILLIR1) : SAVE 200 ADDITION RALITIES. R4, +-RRILLIR1) : SAVE 200 ADDITION RALITIES. R5, +-RRILLIR1) : SAVE 200 ADDITION RALITIES. R6, 172 OF UPPER BATTERCLY. R170 LOPE SRIES. R6,	5 S	##R1++(IRO)B, ##R2++ R4, #R4 LAST_INSIDE_LOOP	(IRO)B, WO (IRO)B, WO I IST NIS LOOP COPPLETE? I INDA PRESENT COUNTER.
R3, R1, R3 + AND 200 SERVATION NUTTINE AND 200 SERVATION NUTTINE (R4, AR4 LAST_INSIDE_LOOP	IS THIS LOOP COMPLETE? DELAYED BROWCH, IF NOT. SET UP REPEAT COUNTER.
R3,R1,R3 ; AND 2ND SUBTRACTION MLTIPLY. R2,+-RR1(IR1) ; SAME IST ANDITION MLTIPLY. ; 1/2 OF BUTTSPELY. TTERELY STAGE LOOP. 11/2 OF LPPER BUTTSPELY.			: DELAYED BRANCH, IF NOT.
R2 +-AR1(IR1) ; SAME IST ADDITION MALTIPLY. R2 +-AR3(IR1) ; SAME DAD ADDITION MALTIPLY I TTERFLY STAGE LOOP. S DOES ALL OF THE BIT REVERSE ADDITIONS AT THE BI N. 2.180 : INVITABLE EDURES ADDITIONS AT THE BI N.			SET UP REPEAT COUNTER.
R2, 4-4R1(IR1) ; SAME IST ADDITION MALTHEY I ; 1/2 OF BUTTERFLY. R2, 4-4R3(IR1) ; SAME 2ND ADDITION MALTHEY I ; 1/2 OF UPPER BUTTERFLY. ITTERFLY STARE LOOP. STATE OF THE BIT REVERSE ANDITIONS AT THE EN W		ARS, RC	
; 1/2 OF UPPER BUTTERLY. TTERLY STACE LOOP. TITION LOOP SERIES. SE DOES ALL OF THE BIT REVERSE ANDITIONS AT THE BIT. THOU IN THE BIT REVERSE ANDITIONS AT THE BIT. THE BIT REVERSE ANDIT	וא טייביא	 	
R3,+-4R3(IR1); SAME 200 ADDITION MLIPLY I; 1/2 OF UPPER BUTTERFLY. TITERFLY STACE LOOP. 11TION LOOP SERIES. 15 DOES ALL OF THE BIT REVERSE ANDITIONS AT THE EN	*	0100H,ST	; SET REPEAT MODE.
RS, +-ARS(IRI); SAME 2ND ANDITION MLIPRY I; 1/2 OF UPPER BUTTERFLY. TITERELY STAGE LOOP. DITTOM LOOP SERIES. SE DOES ALL OF THE BIT REVERSE ANDITIONS AT THE EN TH.	* BRANCH DELAYE	BRANCH DELAYED TO LAST_INSIDE_LOOP.	
TITEMELY STACE LOOP. TITEMELY STACE LOOP. SENDES ALL OF THE BIT REVERSE ANDITIONS AT THE BY	* 03001	VOC B TOO	SO COMEN COO MY SON SCIENT SUNTS .
JITEMELY STAGE LOOP. DITION LOOP SERIES. ES DOES ALL OF THE BIT REVENSE AND TITONS AT THE EN PM. 2. 180 : INITIALIZE THOSE REGISTERS.	•	##R1, ##R2↔(IR1)%, R0	; SINCE INCIDENTE HE HAN UND NOTICE OF 0 ADDITIONS, THE FINAL DNES ARE : DONE NOW.
NITION LODP SERIES. S. DOES ALL OF THE BIT REVENSE ANDITIONS AT THE EN W.	1,607 B 00%		-
SS DOES ALL OF THE BIT REVENSE ANDITIONS AT THE EN. W	- Paracer.		
SS DOCS ALL OF THE BIT REVENSE ADDITIONS AT THE EN M. 2. JRO : INITIALIZE IMOEX REGISTERS	STF	RO, +AR1++(IR1)I	; SAVE ADDITION.
	0 OF FAST * BND OF LAST REPEAT BLOCK.	EPEAT BLOCK.	
	•	1, 180	; MULTIPLY IRO BY 2.
ARE, ARE POINTERS FOR FINAL ADDITION	A001	IRO, R4	. UPDTE INGER LOOP CONTROL REGISTER.
		Q	, ARE CALCULATIONS COMPLETE?
AR1, AR2		LAST_OUTSIDE_LOOP	DELAYED BRANCH, IF NOT.
8, IRI	19	R4, 482	; UPDATE DATA POINTERS.
		1, IR1	; MULTIPLY IRI BY 2.
Compressor that commercial answers	•	THE PERSON TO A PART OF PERSONS AND PARTY OF THE PERSONS AND PARTY OF T	

END OF LAST LOOP SERIES.
MLTIPLY COEFFICIENT ZERO BY .5, 1F NOT ZERO.

DELAYED BRANCH FROM HERE IF VALUE IS NOT TO BE STORED.

ARS, #AR6, AR1

AR1, #AR6

STI

; STORE, IF EXPONENT MASN'T -128.

DONT_STORE: * RETS

Appendix E2. A Fast Cosine Transform (Inverse Transform)

- 1			ď	**************************************	
APPENDIX E2			ADDF3	#4R1++(IRO)B, #4R2++(IRO)B, R0	IRO)B, RO ; FIND FIRST SUM. (MAKES
			5	ARI, AR3	HIDDLE LOOP NORE EFFICIENT)
A FAST COSINE	A FAST COSINE TRANSFORM (INVERSE TRANSFORM)	DANSFORM)	ā	AR2, AR4	
			101	A61 A65	
BASED ON THE	ALCORITHM OUTLINED BY	BASED ON THE ALGORITHM OUTLINED BY BYEONG GI LEE IN HIS ARTICLE, FCT - A	ADDF3	*AR3++(1R0)B. *AR4++	+AR3++(1R0)B +AR4++(1R0)B R1 . DUMMY AND TO UPDATE
FAST COSINE 1	RANSFORM, PUBLISHED IN	FAST COSINE TRANSFORM, PUBLISHED IN THE PROCEEDINGS OF THE IEEE Inter-			POINTERS.
NATIONAL CONF	ENENCE ON ACQUISTICS, S	NATIONAL CONFERENCE ON ACCUSTICS, SPEECH, AND SIGNAL PROCESSING, SAN	3	-1. IRO	: UPDATE INDEX REGISTER.
1 DIEGO, CA, 19	-21 MMCH 1984, P 28A.	DIEGO, CA, 19-21 NARCH 1984, P 28A.3/1-4 VOL 2., (CH1954-5/84/0000-0299).	•	-	
			RPTB	END_CENTER	; TOP OF INNER MOST LOOP.
· LEE'S ALCORIT	HE HAS BEEN HODIFIED 1	LEE'S ALGORITHM HAS BEEN MODIFIED TO ALLON NATURAL ORDER TIME DOMAIN	-		
COEFFICIENTS.			MIDDLE:		; TOP OF MIDDLE LOOP.
THE ERECHENCY	DOMAIN COFFEEDITING A	THE ERECUENCY DOMAIN CREETILIENIS ARE IN BIT REVERSE DRIVER. THIS IS AN IN		2	TO A A COURT OF THE PARTY OF TH
PLACE CALCULATION.	TION			-40° -400° (100° 0° 0°	; UK! UPPER HELF UP SELUND AUDITION.
		-		THERT, THE CAT LINGUIS, IN	
AUTHOR: PAUL WILHELM	WILHELM		- ic	RU, *HRI ** (IRU) B	STORE HULLION DURE THE LIST LOOP ON
			EMO_CENTER:		
ladolg.	IFCT	; INVERSE FAST COSINE TRANSFORM ENTRY	•		
		; POINT.	ADDF3	R3. +AR4++(1R0)B.R0	R3,+AR4++(IRO)B,R0 : D0 SECOND ADDITION.
ladole.		; LENGTH OF ARRAY TO BE TRANSFORMED.	:: STF	RI, *AR3++(IRO)B	STORE FIRST ADDITION.
fadole.		; TABLE OF COSINE COEFFICIENTS.	•		
ladole.	COS_TAB	TABLE OF AMMAY DATA TO BE	* END OF INNER HOST LOOP.	MOST LOOP.	
		; TRONGS-UNDED.	*		
			ADDF3	#AR3++(IR1)X,#AR4++(IR1)X,R2	(IR1)%,R2 ; DUMMY AND TO UPDATE
.text			•		; POINTERS.
	•		Š	+AR3++(JR0)B,R3	; CET VALUE FOR LAST ADDITION.
			<u> </u>	#AP2++(IR0)B,R2	; DUMIY ADD TO UPDATE POINTER.
DAIA . word	136. 136.		ADDF3	R3, +AR4++(1R0)B,R0	; DO LAST ADDITION.
one.	- SD		:: STF	RO, #4R1 ++ (IRO) B	; STORE NEXT TO LAST ADDITION.
1607			ADDF3	*AR1++(IR1)%, *AR2++	+AR1++(JR1)%, +AR2++(JR1)%, R2 ; DUMNY ADD TO UPDATE
	06CTS17F 080	. I nan aasay SIZE.	•		; POINTERS.
3 5	ACTO TE DV	LOAD BLOCK CITE FOR CIDELLAS	ë	180,8C	; UPDATE REPEAT COUNTER.
i	deligite, m	Ampreeting	I de la	ARI, ARS	; IS MIDDLE LOOP COMPLETE?
	A DATA ABL	. I DAN BOINTED TO DATA TABLE		MIDDLE	; IF NOT, DO DELAYED BROWCH.
3 5	P ONC ADT	LOAD BOTWIED TO COLINE TABLE	कु	.,ec	
į	, MOV ADV	DOINT TO LACT CHOINE UNITE THE TABLE	Suer	2,RC	
	7 A07	TOTAL TO CAST COSTAC MACKE IN TROCK.	85	0100H, ST	; SET REPEAT MODE.
1 1 1	, m, 7	. INITIALITY TANKY DEGICTEDS END BIT	•		; (START/STOP ADDRESSES STILL OK)
3 3	ont, the	POLICECT AND TION CONTINUE	•		
5 5	2,110	CARACT MODILION SCHOOLS	+ DELAY BRANCH	DELAY BRANCH FROM HERE TO MIDDLE.	
ij	1996, 4901	. INITIALIZE DATA POINTERS.	•		
900	180.481		5	.180	1 IS OUTSIDE LOOP COMPLETE ?
			9610	OUTSIDE	; IF NOT, DO DELAYED BRONCH.
START OF RIT	START OF BIT REVERSED ADDITION LODP SERIES.		3	AR6, AR1	; PREPARE TO UPDATE POINTERS AT TOP OF
			•		: 100 6 .
OUTSIDE:		: TOP OF OUTSIDE LOOP FOR BIT REVERSED	Abo.	180,481	
		. A00111045.	<u> </u>	-1, IRI	; UPDATE INDEX REGISTER.
1004	180,481	; UPDATE DATA POINTERS AND REPEAT	* DELAY BRONCH	DELAY BRANCH FROM HERE TO OUTSIDE.	
	8	: coorier.	•		
3 5	180, RC		+ END OF BIT RE	END OF BIT REVERSED ADDITION LOOP SERIES.	SENIES.
1975	2.80		-		
į	¥.,'		 START OF CENT 	START OF CENTER BUTTERFLY LOOP.	

STACE, THE FIRST	BUTTERFLY, AND THE COSINE MILTIPLICATIONS FOR THE SECOND BUTTERFLY	
STAGE,	SECOND	
	¥ 85	
THIS LOOP INCLUDES THE LAST BIT REVERSED ADDITION	ATIONS F	
BIT REV	LTIPLIC	
Æ LAST	OSINE N	
L SOUL	6 1 C	
LOOP 1NC	RLY, A	ŝ
HIS.	E	SERIES

1000. 1009. 1000.		ALCIDIES HE LAST BUT TO	INTO LOUP INCLOURS THE CAST BIT REVENSED AUDITION STARK, THE FIRST	.	PI ANDALL
\$181 3,442 INTITALIZE INCENTRATE OF THIS LOPP. 131 4.10 134 1.10 135 4.10 136 4.10 137 5.10 137 6.10 138 1.10 138 1.10 138 1.10 138 1.10 139 1.10 139 1.10 130 1.10	SERIES.		Transition and the second political in	::	R4. +AR3++
SIBI 3.462 I UPMITE INTER RECEIPED. 110				•	
8, JRT. 9, JRT. 11, 14.6. 12, 0.6. 13, 0.6. 11, 14.6. 11, 0.6. 11, 0.6. 11, 0.6. 11, 0.6. 11, 0.6. 11, 0.6. 11, 0.6. 11, 0.6. 12, 0.6. 13, 0.6. 14, 0.6. 14, 0.6. 15, 0.6. 16, 0.6. 17, 0.6. 18, 0.	1875	3,482	; UPDATE DATA POINTER FOR THIS LOOP.	+ DE OF CENT	DIED OF CENTER BUTTERFLY LA
##O ##O ##C ##O ##O ##C ##O ##O ##C ##O ##O ##O ##C ##O ##O ##O ##O ##O ##O ##O ##O ##O ##	Ē	8, IR1	INITIALIZE INDEX REGISTER.	•	
-3. RC, -487, R7 ; GET COSINE P1/4. 11 1. RC, 485 : SAME REPEAT COUNTER FOR LATER USE. 12 1. RC, 485 : SAME REPEAT COUNTER FOR LATER USE. 13 1. RC, 485 : SAME REPEAT COUNTER FOR LATER USE. 14 1. RC, 485 : SAME REPEAT COUNTER FOR LATER USE. 15 1. RC, 482 : SAME REPEAT COUNTER WAS DONE EACH COLOR. 16 1. RC, 487 : SAME REPEAT SAME LONE WAS DONE SAME AND THE COLOR. 17 1. RC, 487 : SAME REPEAT SAME US DONE THE COLOR. 18 1. RC, 481 : RC, 480 : RC, 480 : RC, 481 : RC,	Ē	ARO, RC	: INITIALIZE REPEAT COUNTER.	+ START NEXT	START NEXT TO LAST LOOP SE
### ### ### ### ### ### ### ### ### ##	5	-3,160		•	
1, RC	Ė	**************************************	; OET COSINE P1/4.	* THIS SERIES	MIS SERIES OF LOOPS DOES
### ### ### ### ### ### ### ### ### ##	iens	3,16		+ COSINE COEFF	COSINE COEFFICIENT MILTIP
## BOLCENTER_LIDP FOUR BUTTERFLIES ARE DINE EACH CYCLE	ë	RC, ARS	; SAVE REPEAT COUNTER FOR LATER USE.	+ PLICATIONS P	PLICATIONS FOR THE LAST BL
F3 +4472, 4442, R4 : B11 REVIBEED AND ITON FOR 200 BUTTERFLY. F3 4481, R7 : COSSINE PL/4 TIMES LOARS HALF OF 200 BUTTERFLY. F3 4441, R4 : AND UPPER HALF OF 1ST BUTTERFLY. F3 4441, R5 : BUTTERFLY. F3 R5, +481, R6 : BUTTERFLY. F3 R6, +482, R2 : AND UPPER HALF OF 1ST BUTTERFLY. F3 R6, +482, R2 : AND UPPER HALF OF 1ST BUTTERFLY. F3 R6, +481, R5 : SUBTERFLY. F3 R6, +481, R5 : SUBTERFLY. F4 R7, R2, R0 : MUTTERFLY. F4 R7, R2, R0 : SUBTERFLY. F4 R7, R2, R0 : SUBTERFLY. F4 R87, R2, R0 : SUBTERFLY. F5 R6, +482 : SUBTERFLY. F6 R6, +482 : SUBTERFLY. F7 R63, R64 : R0 : SUBTERFLY. F8 R7, R2, R0 : SUBTERFLY. F9 R7, R4, R3 : SUBTERFLY. F9 R7, R4, R4, R5 : SUBTERFLY. F9 R7, R4, R4, R5 : SUBTERFLY. F9 R7, R4, R6, R1 SUTTERFLY. F9 R7, R4, R6, R1 SUTTERFLY. F1 R4, R6, R1 SUTTERFLY. F1 R4, R6, R1 SUBTERFLY. F1 R4, R6, R1 SUTTERFLY. F1 R4, R6, R1 SUBTERFLY. F1 R4, R6, R1 SUBTERFLY. F1 R4, R7, R7, R1 SUBTERFLY. F1 R4, R1 SUBTERFLY. F1	ET de	END. CENTER 1.00P	FOUR BUTTERS IFS ARE TIME FACH CYCLE	+ FAST EXECUTION.)	ĵ.
##R2, 4#R2, AH : BIT REVERSED ADDITION FOR 200 ##R1, R7, R5 : GOSSINE PL/4 TIMES LOACH NA.F OF 1ST ##TEPETY. ##R4, A-APA, R3 : BIT REVERSED ADDITION FOR 4TH ##TEPETY. ##R4, A-APA, R3 : BIT REVERSED ADDITION FOR 4TH ##TEPETY. ##R4, A-APA, R3 : BIT REVERSED ADDITION FOR 4TH ##TEPETY. ##R5, R6, +481, R4 : GOSSINE PL/4 TIMES LOACH NA.F OF 4TH ##TEPETY. ##R5, R6, +481, R7 : GOSSINE PL/4 TIMES LOACH NA.F OF 4TH ##TEPETY. ##R5, R7, R8 : SAURTHAY LOACH NA.F OF 1ST ##TEPETY. ##R5, R7, R8 : SAURTHAY LOACH NA.F OF 200 BUTTEPETY. ##R5, R7, R4 : GOSSINE PL/4 TIMES LOACH NA.F OF 200 ##TEPETY. ##R5, R7, R4 : GOSSINE PL/4 TIMES LOACH NA.F OF 200 ##TEPETY. ##R5, R7, R4 : GOSSINE PL/4 TIMES LOACH NA.F OF 200 ##TEPETY. ##R5, R7, R4 : GOSSINE PL/4 TIMES LOACH NA.F OF 200 ##TEPETY. ##R5, R7, R4 : GOSSINE PL/4 TIMES LOACH NA.F OF 200 ##TEPETY. ##R5, R7, R4 : GOSSINE PL/4 TIMES LOACH NA.F OF 200 ##R7, R2, R0 : NATITERY LOACH NA.F OF 200 ##R7, R3, R1 : +APA, R3 : SAURHHAY LOACH NA.F OF 200 ##R7, R3, R1 : +APA, R3 : SAURHHAY LOACH NA.F OF 200 ##R7, R3, R1 : +APA, R3 : SAURHHAY LOACH NA.F OF 300 ##R7, R3, R1 : +APA, R3 : SAURHHAY LOACH NA.F OF 300 ##R7, R3, R1 : +APA, R3 : SAURHHAY LOACH NA.F OF 300 ##R7, R3, R1 : +APA, R3 : SAURHHAY LOACH NA.F OF 300 ##R7, R3, R1 : +APA, R3 : SAURHHAY LOACH NA.F OF 300 ##R7, R3, R4, R4, R5 : SAURHHAY LOACH NA.F OF 300 ##R7, R3, R4, R4, R5 : SAURHHAY LOACH NA.F OF 300 ##R7, R8, +APA, R8, R8, R9, R9, R9, R9, R9, R9, R9, R9, R9, R9			THROUGH THIS LOOP.	iens	2.487
F3 +4472,4442, P4 1811 REMBER ANDITION FOR 200 BUTTERFY. F3 +481,77,85 1.00518E P1/4 TIMES LOMEN HALF OF 310 BUTTERFY. F3 +481,77,87 1811 REMBERS ANDITION FOR HALF F3 +481,782,181 1811 REMBERS ANDITION FOR HALF F3 +4817,782,781 1811 REMBERS ANDITION FOR HALF F3 +4817,782,781 1811 REMBERS ANDITION FOR HALF F3 +4817,782,781 1811 REMBERS AND LEMBER HALF F3 +4817,782,781 1811 REMBERS HALF F3 +4817,782,781 1811 REMBERS HALF F3 +4817,782,781 1811 REMBER HALF F3 +4817,782,781 1811 HALTIPHY LOWER HALF F5 +4817,782,781 1811 HALTIPHY F5 +4817,782,781 1811 HALTIPHY F5 +4817,782,781 1811 HALTIPHY F5 +4817,782,781 1811 HALTIPHY F5 +4817,782,782 1817,7431 18					1.484
### #### ### ### ### ### ### ### ### #	ADDF3	*** ##2, **#2, R4	; BIT REVERSED ADDITION FOR 2ND	9	5.6
## ## ## ## ## ## ## ## ## ## ## ## ##			; DUTERPLY.	5	##77- R5
### ##################################	MPYF3	*#R1,R7,R5	; COSINE PI/4 TIMES LOWER HALF OF 1ST	Ė	#4R7, R4
### ### ### ### ### ### ### ### ### ##	į	:	BUTTERELY.	•	
F3 + 4444, R3 19 IT REVENSED AND ITION FOR 4TH BUTTERFY. F3 15, +441, R4 400 UPPEN HALF OF 1ST BUTTERFY. F3 16, +441, R4 400 UPPEN HALF OF STD BUTTERFY. F3 16, +441, R5 10 SSIR E PL14 TIMES LUBER HALF OF 4TH BUTTERFY. F3 16, +441, R5 10 SSIR E PL14 TIMES LUBER HALF OF 200 BUTTERFY. F3 16, +441, R5 10 SSIR CAPETIBINT. F3 16, +442, R2 10 SSIR CAPETIBINT. F3 16, +442 10 SSIR E PL14 TIMES LUBER HALF OF 200 BUTTERFY. F467, R2, R0 10 SSIR E PL14 TIMES LUBER HALF OF 200 BUTTERFY. F467, R2, R0 10 SSIR E PL14 TIMES LUBER HALF OF 200 BUTTERFY. F467, R2, R0 10 SSIR E PL14 TIMES LUBER HALF OF 200 BUTTERFY. F467, R3, R1 10 SSIR CAPETICIENT F467, R3, R4 10 SSIR C		K/, K4, K0	; COSINE P1/4 TIMES LOWER HALF OF 2ND	eT96	END_IMIL
### 1 ### 1	AMPES	20 404 4 404	pur polymers appring on an	•	
F3 R3, +481, R4 i ADD UPPER HAF OF 1ST BATTBRELY. F3 R0, 4462, R2 i GOSSIE PLY LT TIMES LOAR HAF OF 4TH BATTBRELY. F3 R0, 4462, R2 i ADD UPPER HAF OF 25D BATTBRELY. F3 R0, 4462, R2 i BATTBRELY. F3 R0, 4462, R2 i BATTBRELY. F4 R4 HA I STORE UPPER HAF OF 25D BATTBRELY. F4 R4 HA I STORE UPPER HAF OF 25D BATTBRELY. F4 R4 HA I STORE UPPER HAF OF 25D BATTBRELY. F4 R4 HA I STORE UPPER HAF OF 25D BATTBRELY. F5 R4 HA I STORE UPPER HAF OF 25D BATTBRELY. F6 R4 HA I STORE UPPER HAF OF 25D BATTBRELY. F6 R4 HA R7 STORE UPPER HAF OF 25D BATTBRELY. F7 R4 HA R7 STORE UPPER HAF OF 37D BATTBRELY. F8 R4 HA R7 STORE UPPER HAF OF 37D BATTBRELY. F8 R4 HA R7 STORE UPPER HAF OF 37D F7 SAD UPPE		apere, a years, M.S	t bit reverses restitute for 4th	• !	
10.0518 F 174 TIMES LORER HALE OF 4TH 10.0518 C 174 TIMES LORER HALE OF 151 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMES LORER HALE OF 200 BUTTESTLY 10.0000 F 175 TIMESTLY 10.00000 F 175 TIMESTLY 10.000000 F 175 TIMESTLY 10.000000 F 175 TIMESTLY 10.0000000 F 175 TIMESTLY 10.00000000 F 175 TIMESTLY 10.00000000000000000000000000	ANNES	70 TOV-1 NO	AND INDER LAW F. OF ACT PARTITION A	MILLOOP:	
### ### ### ### ### ### ### ### ### ##	PYF3	###7 R3 R1	CONSTRUCTION THE OF 151 BUILDING.		1000
53 60, 4482, R A AD UPPER MAY OF 200 BUTTERELY			: BUTTERELY.		PIRCH, PRICS, P
F3 16, +-481, R5 ; SURTINGT LOAGE HALF OF SST MUTTENT LY OFFER HALF OF 200 BUTTENT LY R0, +462, R2 ; SURTINGT LOAGE HALF OF 200 BUTTENT LY R0, +462, R2 ; SURTINGT LOAGE HALF OF 200 BUTTENT LY R0, +462, R1 ; STONE LOAGE HALF OF 200 BUTTENT LY R0, +464, H181, STONE LOAGE HALF OF 200 BUTTENT LY R0, +464, R3 ; SURTENT LOAGE HALF OF 200 BUTTENT LY R1, +464, R3 ; SURTINGT LOAGE HALF OF 200 BUTTENT LY R1, +464, R3 ; SURTINGT LOAGE HALF OF 200 BUTTENT LY R1, +464, R3 ; SURTINGT LOAGE HALF OF 200 BUTTENT LY R1, +464, R3 ; SURTINGT LOAGE HALF OF 200 BUTTENT LY R1, +464, R3 ; SURTINGT LOAGE HALF OF 300 R1, H184, R3 ; SURTINGT LOAGE HALF OF 300 R1, +464, R3 ; SURTINGT LOAGE LOAG	ADDF3	RO, #4R2, R2	: ADD UPPER HALF OF 2ND BUTTERFLY.	AMDE:	0 0004 1004
### ### ### ### ### ### ### ### ### ##	SUBF3	75,+-181,R5	; SUBTRACT LOWER HALF OF 1ST	E MANES	8
### ### ### ### ### ### ### ### ### ##			; BUTTERPLY.	•	21112
Pr OSSING COEFFICIENT	EVE3	+-4R7, R2, R0	; MILTIPLY UPPER HALF OF 200 BUTTERFLY	:: ADDF3	#MC2. #MR1. R
53 50, 4462, R2 18.28	į		; BY COSINE COEFFIEIENT.	MPVF3	R4. R7. R1
M. +ARI M. +ARI M. +ARI M. TOBE LUMER HALF OF 1ST BUTTERFU. M. +ARZ STONE LUMER HALF OF STON BUTTERFU. M. +ARZ STONE LUMER HALF OF 200 BUTTERFU. M. +ARZ M. TOBEL LUMER HALF OF 200 BUTTERFU. M. TIPRY LUMER HALF OF 200 BUTTERFU. M. HARPAR, R. MASSINGN LUMER HALF OF 300 M. +ARZ, R. M. M. M. TIPRY LUMER HALF OF 41H M. +ARZ, R. M. M. M. TIPRY LUMER HALF OF 41H M. +ARZ, R. M.	STREET	RO, #MR2, R2	; SUBTRACT LOKER HALF OF 2ND	•	
M. +-M. 1 1006 UPM MA D 151 BUTBELY. M. +-M. 1 1006 UPM MA D 151 BUTBELY. M. +-M. 1 1006 UPM MA D 151 BUTBELY. M. +-M. 1 1006 UPM MA D 10 20 BUTBELY. M. +-M. 1 1006 UPM MA D 0 20 BUTBELY.	į	2	; BUTTEMELY.	:: SUBE3	+4R2, +4R1, R
10.4482 10.00 LIMEN HAL OF 151 DIDENTY. 483.78.7,44 COSINE PL/A TIMES LOAD BUTTENTY. 483.78.7,74 COSINE PL/A TIMES LOAD BUTTENTY. 53	<u>,</u>	184, 4-481	; STONE UPPER HALF OF IST BUTTERFLY.		
73 447, 784 1 COSINE THAT DE LOAD BUTTERELY. 74 487, 784 1 COSINE THAT DE LOAD BUTTERELY. 75 447, 78, 10 STEPLEN HALF OF 200 BUTTERELY. 75 447, 78 1 STEPLEN HALF OF 200 BUTTERELY. 75 7 1 STEPLEN HALF OF 300 BUTTERELY. 75 7 1 STEPLEN HALF OF 300 BUTTERELY. 76 7 1 STEPLEN HALF OF 300 BUTTERELY. 77 7 1 STEPLEN HALF OF 300 BUTTERELY. 78 7 1 STEPLEN HALF OF 300 BUTTERELY. 79 80 1 STEPLEN HALF OF 300 BUTTERELY. 79 80 1 STEPLEN HALF OF 300 BUTTERELY. 70 80 1 STEPLEN HALF OF 300 BUTTERELY. 70 80 1 STEPLEN HALF OF 300 BUTTERELY. 71 80 1 STEPLEN HALF OF 300 BUTTERELY. 72 80 1 STEPLEN HALF OF 300 BUTTERELY. 73 80 1 STEPLEN HALF OF 300 BUTTERELY. 74 80 1 STEPLEN HALF OF 300 BUTTERELY. 75 80 1 STEPLEN HALF OF 300 BUTTERELY.	<u></u>	FD, #48(1++ (1R1))	STONE LONER HALF OF 1ST BUTTERFLY.	STF	RO, #AR3++(
###O.187.197 MOLINE PLY TIMES LOBER HAFF OF 300 ###O.182.190 MLITIEV LOBER HAFF OF 200 BUTTEPELY ###O.182 W. O.181 CORFFICIENT ###O.183 W. O.181 CORFFICIENT ###O.183 W. O.181 CORFFICIENT ###O.183 W. O.181 CORFFICIENT ###O.183 W. MLITIEV LOBER HAFF OF 41H BUTTEPELY ###O.183 W. MLITIEV LOBER HAFF OF 41H BUTTEPELY ###O.183 W.	<u>.</u>	10, e+802	; STUME LOWER HALF OF 2ND BUTTERFLY.	:: STF	R2, #4R1++(1
F3 ***7, R2, R0 ***MLTIPLY LOUGH MALF OF 200 BUTTERFLY F3 R1, +**44, R3 ***SUBTECT LOUGH MALF OF 200 BUTTERFLY F3 R4, +**483, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F3 ***487, R3, R1 ***MLTIPLY LOUGH MALF OF 3400 BUTTERFLY F3 ***487, R3, R1 ***MLTIPLY LOUGH MALF OF 3400 BUTTERFLY F3 R1, +**484, R3 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F4 ***MR3, R4 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F4 ***MR3, R4 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F4 ***MR3, R4 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F4 ***MR3, R4 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F4 ***MR3, R4 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F4 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F4 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F4 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F4 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F4 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFLY F5 ***MR3, R5 ***SUBTECT LOUGH MALF OF 3400 BUTTERFL	2	March, N., Ne	COSTRE PL/4 LITES LUNER HALL UF 340	• ·	
F3 R1,+-484,R3 F3 R4,+-482,R5 F3 R1,+-684,R3 F3 R4,+-482,R4 F3 +-487,R3,R1 R1,+-484 R1,+-484 R5,+-484	MPYF3	*AR7. R2. R0	. Mailley Louis Hair Of Maillean V		
F3 R4, +-484, R3 F3 F4, +-487, R5 F3 R4, +-487, R3, R1 F3 R4, +-487, R4 F3 F4			BY COSINE COEFFICIENT	•	
33 RA,+-480,R5 13 84,-480,R3 13 RA,+-280,R4 13 +-487,R3,R4 13 +-487,R3,R1 15 +-487,R3,R1 16 +-487	SUBF3	RI, +-4R4, R3	SUBTRACT LOWER HALF OF 4TH	÷ 5	K1, 498(4++(1
7.3 444.7463.55 7.3 4467.763.81 7.3 R1.+2464.83 7.3 R4.+2463.84 7.3 +2467.83.81 7.3 +2464 7.5 +2464 7.5 +2464			; BUTTERELY.	- •	13, mar.cm
73 ************************************	ADDF3	R4, +-483, R5	; ADD UPPER HALF OF 3RD BUTTER-LY.	+ FIN OF CENTE	PAN OF CENTER LONG OF MEYT
73 R1,+484,R3 73 R4,+483,R4 73 +-487,R3,R1 73 +-487,R3,R1 74 +-484 75 +-484 75 +-487 75 +-487	EME3	##87, R3, R1	; MULTIPLY ; ONER HALF OF 4TH BUTTERFLY	•	
7-3 R4, +-880, R4 ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	***************************************		# BY COSINE COEFFICIENT	ij	35. RC
73 #-#87,R3,R1 R1,#-484 R0,#482+(1R1)X R5,+487		K1, 1984, K3	; ADD UPPER HALF OF 4TH BUTTERFLY.	Š	#4R7 R5
F3 +-487, R3, R1 ; ; R1, +-484 ; ; R0, +482+(IR1) x ; ; R4, +482+(IR1) x ; ; R5, +-487	7	#*, **	* SUBTRACT LOADS HALF OF 340	Ė	**************************************
R1, +-AR4 R0,+M2+(R1)X R5,+-AR2	HPYF3	4-487. R3. R1	* BOLLEGELY. • MELTIPLY LIPPER HALF OF ATH BUTTERS! V	*	
R1,+-4R4 R0,+4R2++(JR1)X R5 +-4R3			BY COSINE COEFFICIENT.	-	į
RO,+482+(IR1)X	STF	R1, #-4R4	STORE UPPER HALF OF 4TH BUTTERRLY.	5	98.
29.	STF	RO, +AR2++(IR1)X	STORE UPPER HALF OF 2ND BUTTERFLY.		NIC.LOOP
2	STF	75, 1 183	STORE UPPER HALF OF 380 BUTTERELY.	ADDF.3	##### ***

* DIOLODITER LOOP: *

SET REPEAT MODE. (START/STOP ADDRESSES ARE STILL GOOD.)	UPDATE DATA POTATERS. UPDATE INDEX REGISTER. IS THIS LODE SERIES COMPLETE? IF NOT, REMACH RELAYED. UPDATE DATA POTATER. UPDATE REPEAT COUNTER.	DELAYED BRANCH FROM HERE TO INTLLODP. 300 OF HEXT TO LAST LOOP SERIES. START OF THE LAST LOOP. START OF THE LOST WITHOUT THE COSINE CIEFFICIENT MALTHRUSTING, MATCH HAVE ALGEBY BEEN DONE.	INITIALIZE INDEX REGISTER. INITIALIZE DRIA POINTERS. INITIALIZE REPEAT COUNTER.	≅ ⊌ इस्त	BUTTBEFLY. STORE LIPER HALF OF 200 BUTTBEFLY. STORE LIPER HALF OF 200 BUTTBEFLY. STORE LOARN HALF OF 200 STORE LOARN HALF OF 200 STORE LOARN HALF OF 200 BUTTBEFLY. STORE LOARN HALF OF 200 BUTTBEFLY.	; STONE LOWER HALF OF 2ND BUTTENELY.
0R 0100H, ST ::	MANACH DELAY FROM HERE TO NITLLOOP. LOT AND 181, AR1, AR3, AR1 LSH 1,1R1 OPP 1 181, AR0 ADD 13 1R0, AR3, AR4 LSH 1,1R5 ADD 13 1R0, AR3, AR4 LSH 1,1R5 LSH 7,8R5 LSH 7,8R5 LSH 7,8R5	DELAYED BRANCH FROM HERE TO NTLLLOOP, BOD OF NEXT TO LAST LOOP SERIES. START OF THE LAST LOOP. THE LAST LOOP IS THE LAST BUTTERELY S MALTIPLICATIONS, WHICH HAVE ALPEADY E	LDI 2,181 ANDI3 1RO,462,684 ; SUB13 1RO,462,684 ; SUB13 1RO,462,684 ; LDI 4800,RC ; LSH -2,RC ;	#### BIBLIASTILOOP ; LIDF ####2, ###1, R1 ; SUBF3 ###2, ###1, R2 ;	STF R1, 4481—(1R1) ; SUBF3 R0, 4482,R8 ; STF R2, 4481—(1R1) ; STF R2, 4482—(1R1) ; STF R2, 4483—(1R1) ;	R4, +684++(IR1)
4 8		DELAYED BIO OF 1 START OF THE LAS'	20 A A A A A A A A A A A A A A A A A A A	9	ল'ন ফন≇	STF STF

BID OF LAST LOOP, AND INVERSE COSINE TRANSFORM.

end.

Appendix E3. FCT Cosine Tables File

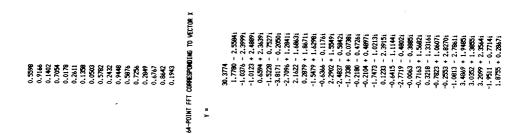
```
¥
¥
     APPENDIX E3
¥
¥
     FOT COSINE TABLES FILE
¥
¥
     TO BE LINKED WITH FCT SOURCE CODE FOR 32 POINT FCT.
¥
     COEFFICIENTS ARE 1/(2 * COS(N*PI/2M)), WHERE N IS A NUMBER FROM 1 to
¥
     M-1. M IS THE ORDER OF THE TRANSFORM.
¥
¥
¥
     FOR A 32 POINT FCT, N IS IN THE FOLLOWING ORDER: 1
          1, 15, 3, 13, 5, 11, 7, 9,
¥
¥
          2, 14, 6, 10,
¥
          4, 12,
¥
          8
¥
*
     THE LAST VALUE IN THE TABLE IS 2/M.
¥
¥
          .global
                    COS_TAB
          .global
                    M
¥
М
          .set
                    16
¥
          .data
COS_TAB
          .float 0.5024193
          .float 5.1011487
.float 0.5224986
         .float 1.7224471
.float 0.5669440
          .float 1.0606777
          .float
                  0.6468218
          .float 0.7881546
          .float 0.5097956
          .float
                  2.5629154
          .float 0.6013449
          .float
                  0.8999762
         .float 0.5411961
         .float 1.3065630
.float 0.7071068
          .float
                    0.1250000
          .end
```

Appendix E4. Data File

```
Ä
     APPENDIX E4
¥
¥
     DATA FILE
¥
¥
                     COEFF
          .global
          .data
COEFF
          .float
                     137.0
          .float
                     249.0
          .float
                     105.0
          .float
                     217.0
                     73.0
          .float
          .float
                     185.0
                     41.0
          .float
          .float
                     153.0
           .float
                     9.0
                     121.0
          .float
           .float
                     233.0
           .float
                     89.0
                     201.0
           .float
           .float
                     57.0
           .float
                     169.0
           .float
                     25.0
           .end
```

Appendix F. Test Vectors, 64-Point Sine Table, Link Command File

Appendix F1. Example of a 64-Point Vector to Test the FFT Routines

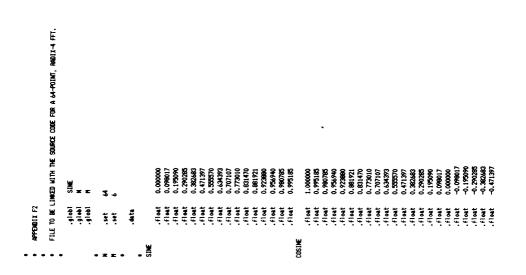


0.0087 0.0087 0.0087 0.0087 0.0087 0.0087 0.0087 0.0087 0.0088 0.

-1.5974 -1.5914 0.7714 3.029 - 2.3844 3.029 - 2.3844 3.022 - 1.3651 -1.0613 + 2.7861 -0.782 - 1.0607 -0.782 - 1.0607 -0.782 - 1.0607 -0.782 - 1.0607 -0.782 - 1.0607 -0.782 - 1.1945 -0.782 - 1.2867 -0.084 - 0.3867 -0.084 - 0.3867 -0.084 - 0.3867 -0.2109 - 0.4924 -0.2209 - 0.2394 -0.2309 - 0.2394 -0.2309 - 0.2394 -0.2309 - 0.2394 -0.2309 - 0.2394 -0.2304 - 0.2394 -0.2306 - 0.2394 -0.2306 - 0.2394 -0.2306 - 0.2394

Appendix F2. File to Be Linked with the Source Code for a 64-Point, Radix-4 FFT.





Appendix F3. Link Command File

```
*
    APPENDIX F3

*
    LINK COMMAND FILE

*
    DO NOT TYPE IN THESE FIRST SEVEN LINES
-0 12opt64.out
12fopt.obj
sin64.obj

SECTIONS
{
    .text : {}
    .data : {}
    IN 809800h : { 12fopt.obj(IN) }
    .bss 809C00h: {}
}
```