DECUS
PROGRAM LIBRARY

DECUS NO.	8-260
TITLE	TOFAST-FAST DIRECT AND INVERSE DISCRETE FOURIER TRANSFORM ROUTINES
AUTHOR	Peter L. Walton Submitted by Dr. William S. Yamamoto
COMPANY	School of Medicine University of Pennsylvania Philadelphia, Pennsylvania
DATE	January 26, 1970
SOURCE LANGUAGE	MACRO-8

1. Program Title: TOFAST - Fast Direct and Inverse Discrete Fourier Transform Routines.

2. Abstract: TOFAST is capable of calculating in place either the discrete Fourier transform (DFT) of real series $\{y_j\}$, $j = \emptyset,1,\ldots,N-1$, where $N = 2^M$ and $M = 3,4,\ldots,1\emptyset$, or the inverse discrete Fourier transform (DFT^{-1}) of the Fourier cosine and sine coefficients, a_k, $k = \emptyset,1,\ldots,n$, and b_k, $k = 1,2,\ldots,n-1$, respectively, where $n = N/2$. Output of the DFT is the Fourier cosine and sine coefficients; output of the DFT^{-1} is the real series. Several modifications are possible which allow for input and/or output to be complex-valued.

3. Hardware Requirement: PDP-8 with minimal of 8K of core memory. (EAE is optional depending on version of floating-point package used (see Item 4)).

4. Software Requirement: One of the following extended memory floating-point packages must be in the memory field containing TOFAST, and must be able to be called by a JMS I 7: FPPES4, FPPES5, FPPES6, FPPEH4, FPPEH5, or FPPEH6. (The last three require EAE).

5. Core Limits: TOFAST uses
   Field $\emptyset$ locations: $2\emptyset$-37, 64-77, $2\emptyset\emptyset$-277, $4\emptyset\emptyset$-1377;
   Field 1 locations: $2\emptyset\emptyset$-(3·N+177), N is as in Item 2.
   Field $\emptyset$ locations 25-37, 64-77 are used for variables and thus are available for that use by co-existing but independent programs defined by the user.

6. Loading Procedure:
   (a) Load the extended memory floating-point package.
   (b) Load TOFAST binary tape.

7. Starting Addresses:
   (a) For the DFT, SA is Field $\emptyset,\emptyset4\emptyset\emptyset$.
   (b) For the DFT^{-1}, SA is Field $\emptyset,\emptyset4\emptyset1$

8. Use: Prior to entry for either the DFT or the DFT^{-1}, the data array beginning at Field 1, $\emptyset2\emptyset\emptyset$ must be properly filled:
   (a) For the DFT, the real series $\{y_j\}$ is stored in j-sequence and without gaps, i.e. $y_\emptyset, y_1, \ldots, y_{N-1}$. The data format can be either signed, 11-bit fixed-point, or standard floating-point. (Standard floating-point format refers to that described in the Floating-Point System Programming Manual, Digital 8-5-S).
   (b) For the DFT^{-1}, the Fourier cosine and sine coefficients are stored as the sequence $a_\emptyset, a_1, \ldots, a_n, b_1, b_2, \ldots, b_{n-1}$. The format is standard floating-point.

With the data array filled as above, LOAD ADDress ØØØ4ØØ for the DFT, or ØØØ4Ø1 for the DFT^{-1}. Set the switch register (SR) to N = size of data array; that is, for the DFT, N = length of series $\{y_j\}$, and for the DFT^{-1}, N = total number of Fourier cosine and sine coefficients. In either case, N must be of the form $N = 2^M$, where M = 3,4,...,1Ø. Press START. The DFT^{-1} will run to completion, halting at ØØØ551 with C(AC) = Ø. The DFT, however, will halt at ØØØ4Ø3 for format information. If the data format is floating-point, set SR to 4ØØØ; if fixed-point, set SR to number of bits to the right of the binary point. Press CONTinue. The DFT will run to completion and halt at ØØØ444 with C(AC) = Ø.

There is one error halt. It is at ØØØ472 with C(AC) = 7777, and occurs if N is not of an acceptable form. Recovery is possible by resetting the SR to an acceptable N and pressing CONTinue.

9. Operating Procedures: Assuming data array is filled according to Item 8,

(a) for the DFT,

(1) set SR to ØØØ4ØØ;

(2) Press LOAD ADDress;

(3) set SR to N =no. of data in series;

(4) press START;

(5) at halt ØØØ4Ø3, set SR to

4ØØØ if floating-point data,

or

no. of bits right of binary point if fixed-point data;

(6) press CONTinue:

(b) for the DFT^{-1},

(1) set SR to ØØØ4Ø1;

(2) press LOAD ADDress;

(3) set SR to N = total no. of cosine and sine coefficients;

(4) press START.

10. Output: At completion of the DFT, the data array contains the Fourier cosine and sine coefficients, a_k, and b_k, respectively, in the sequence $a_Ø, a_1, \ldots, a_n$, $b_1, b_2, \ldots, b_{n-1}$, and in standard floating-point format, where

$$a_k = \frac{2}{N} \sum_{j=Ø}^{N-1} y_j \cos\left(\frac{2\pi}{N} jk\right) \quad \text{for } k = Ø, 1, \ldots, \frac{N}{2}$$

and

$$b_k = \frac{2}{N} \sum_{j=\emptyset}^{N-1} y_j \sin\left(\frac{2\pi}{N} jk\right) \text{ for } k = 1,2,\ldots,\frac{N}{2}-1$$

and the series y_j that was transformed can be represented by

$$y_j = \frac{a_\emptyset}{2} + \sum_{k=\emptyset}^{\frac{N}{2}-1} \left[a_k \cos\left(\frac{2\pi}{N} jk\right) + b_k \sin\left(\frac{2\pi}{N} jk\right)\right] + \frac{a_{\frac{N}{2}}}{2} \cos(\pi j)$$

for $j = \emptyset,1,\ldots,N-1$.

At completion of the DFT^{-1}, the data array contains the series $\{y_j\}$ for $j = \emptyset,1,\ldots,N-1$, in sequence and in standard floating-point format, and whose Fourier cosine and sine coefficients were contained in the input data array.

11. Discussion: The method employed by TOFAST is a modification of the Cooley-Tukey fast Fourier transform algorithm.

For the DFT, the series $\{y_\ell\}$, $\ell = \emptyset,1,\ldots,N-1$, and $N = 2^M$, $M = 3,4,\ldots,1\emptyset$, is first formed into the complex sequence

$$X_j = y_{2\tilde{j}} + iy_{2\tilde{j}+1} \quad \text{for } j = \emptyset,1,\ldots,n-1$$

where $n = N/2 = 2^m$, and where if the binary representation of j is $j = j_{m-1}j_{m-2}\cdots j_1 j_\emptyset$, then $\tilde{j} = j_\emptyset j_1 \cdots j_{m-2}j_{m-1}$. Letting X^p denote the p-th intermediate vector, and defining $Z(j,k) = j_{m-1}j_{m-2}\cdots j_{k+1}\emptyset j_{k-1}\cdots j_1 j_\emptyset$, $I(j,k) = j_{m-1}j_{m-2}\cdots j_{k+1}1j_{k-1}\cdots j_1 j_\emptyset$, and $L(j,k) = j\cdot 2^k \pmod n$,

let $$X_j^\emptyset = X_j \quad \text{for } j = \emptyset,1,\ldots,n-1,$$

and $$X_j^p = X_{Z(j,p-1)}^{p-1} + X_{I(j,p-1)}^{p-1} \cdot W^{L(j,m-p)}$$

for $j = \emptyset,1,\ldots,n-1$ for every $p = 1,2,\ldots,m$, where $W = e^{-2\pi i/n}$, and $i = \sqrt{-1}$. Then the complex Fourier coefficient $Y(\omega_k)$ at frequency $\omega_k = 2\pi k/n$ is

$$Y(\omega_k) = \begin{cases} \mathrm{Re}[X_\emptyset^m] + \mathrm{Im}[X_\emptyset^m] & \text{for } k = \emptyset \\ \frac{1}{2}[X_k^m + (X_{n-k}^m)^*] + \frac{W^{k/2}}{2i}[X_k^m - (X_{n-k}^m)^*] & \text{for } k = 1,2,\ldots,n-1 \\ \mathrm{Re}[X_\emptyset^m] - \mathrm{Im}[X_\emptyset^m] & \text{for } k = n \end{cases}$$

where $(X^m_{n-k})^*$ is the complex congugate of X^m_{n-k}. Finally

$$a_k = \frac{1}{n}\mathrm{Re}[Y(\omega_k)] \quad \text{for } k = \emptyset,1,\ldots,n,$$

and

$$b_k = -\frac{1}{n}\mathrm{Im}[Y(\omega_k)] \quad \text{for } k = 1,2,\ldots,n-1,$$

where a_k, b_k are the Fourier cosine and sine coefficients, respectively, at frequency $2\pi k/N$.

For the DFT^{-1}, the Fourier cosine and sine coefficients, a_k and b_k, respectively, are formed into the modified complex coefficients

$$\hat{Y}(\omega_k) = \begin{cases} a_\emptyset & \text{for } k = \emptyset \\ a_k - ib_k & \text{for } k = 1,2,\ldots,n-1 \\ a_n & \text{for } k = n \end{cases}$$

where $\hat{Y}(\omega_k) = \frac{1}{n} Y(\omega_k)$. Using the definitions and notations given for the DFT, let

$$V_k = \begin{cases} \frac{1}{2}[\hat{Y}(\omega_\emptyset) + \hat{Y}(\omega_n)] - \frac{1}{2i}[\hat{Y}(\omega_\emptyset) - \hat{Y}(\omega_n)] & \text{for } k = \emptyset \\ \frac{1}{2}[\hat{Y}(\omega_k) + (\hat{Y}(\omega_{n-k}))^*] - \frac{W^{-k/2}}{2i}[\hat{Y}(\omega_k) - (\hat{Y}(\omega_{n-k}))^*] & \text{for } k = 1,2,\ldots,n-1 \end{cases}$$

and $X^\emptyset_j = V_j$ for $j = \emptyset,1,\ldots,n-1$

and

$$X^p_j = X^{p-1}_{Z(j,p-1)} + X^{p-1}_{I(j,p-1)} \cdot W^{-L(j,m-p)}$$

for $j = \emptyset,1,\ldots,n-1$ for every $p = 1,2,\ldots,m$. Finally, the series $\{y_j\}$ is given by

$$y_j = \begin{cases} \mathrm{Re}[X^m_{j/2}] & \text{for } j = \emptyset,2,4,\ldots,N-2 \\ \mathrm{Im}[X^m_{(j-1)/2}] & \text{for } j = 1,3,5,\ldots,N-1 \end{cases}$$

12. Some Possible User Modifications: It may be desired to locate TOFAST in field N and the data array in field M. To do this, the following changes to TOFAST should be made:

REGISTER	FROM	TO	CODING
Ø441	ØØ17	ØN17	FCDF NØØ
Ø517	Ø117	ØM17	FCDF MØØ
Ø546	ØØ17	ØN17	FCDF NØØ
Ø71Ø	6211	62M1	CDF MØ
Ø717	62Ø1	62N1	CDF NØ

In addition, several modifications to the extended memory floating-point package must be made in order to relocate it in field M. (See write-up for Overlay Modifications to the Floating-Point Packages.)

There are four locations from which control can be transferred after the DFT and the DFT^{-1} to a user-defined program; these four locations contain NOP's in the binary tapes supplied. After the DFT, jump from Ø437 retaining the floating data field (FDF), or from Ø443 not retaining the FDF; after the DFT^{-1} jump from Ø544 retaining the FDF, or from Ø55Ø not retaining the FDF.

If the complex Fourier coefficients $Y(\omega_k)$ for the real series $\{y_j\}$ are desired as final output, make the following changes to TOFAST:

REGISTER	FROM	TO	CODING
Ø43Ø	1255	1353	TAD LFNOP
Ø432	1255	1353	TAD LFNOP
Ø434	1255	1352	TAD FGETP5
Ø525	1352	1255	TAD FDIVFN
Ø527	1352	1255	TAD FDIVFN
Ø552	3351	5351	FGETP5, 5351
1234	1Ø26	1Ø3Ø	TAD NN
1264	5351	5361	FGET FN

Use is as in Item 8, with N = length of $\{y_j\}$. Output is the complex Fourier coefficients $Y(\omega_k) = \alpha_k - i\beta_k$, for k = Ø,1,...,n, stored as $\alpha_Ø, \alpha_1, \alpha_2, \ldots, \alpha_n, \beta_1, \beta_2, \ldots, \beta_{n-1}$. If $Y(\omega_k)$ are transformed by the DFT^{-1} with the latter changes and using N = 2n, the output is the series $\{y_j\}$.

If the series $\{Y_j\}$ to be transformed is complex-valued make the following alterations to TOFAST:

REGISTER	FROM	TO	CODING
• Ø412	4616	4617	JMS I SPREAL
Ø413	4617	4754	JMS I CPXPML
Ø422	462Ø	4754	JMS I CPXPML
Ø43Ø	1255	5237	JMP 437
Ø5Ø5	724Ø	7ØØØ	NOP
Ø511	711Ø	7ØØØ	NOP
Ø522	777Ø	7774	M4, -4
Ø523	6ØØ7	7ØØ3	M775, -775
Ø525	1352	5334	JMP534
Ø543	462Ø	7ØØØ	NOP

fill data array with $Y_j = x_j - iy_j$ as $x_\emptyset, x_1, \ldots, x_{N-1}, y_\emptyset, y_1, \ldots, y_{N-1}$. Use is as in Item 8, with N = length of $\{Y_j\}$. Output is the Fourier coefficients $Y(\omega_k) = \alpha_k - i\beta_k$, stored as $\alpha_\emptyset, \alpha_1, \ldots, \alpha_n, \beta_1, \beta_2, \ldots, \beta_{n-1}$. If $Y(\omega_k)$ are transformed by the DFT^{-1} with the latter changes and using N = 2n, the output is the complex series $\{Y_j\}$ stored as above.

13. Logic Flow Charts: The two flow charts below are intended only to show the logic structures of the DFT and the DFT^{-1} of TOFAST; no attempt has been made to parallel in any way the physical structure of TOFAST and the flow charts.

14. Program Listing: Listing follows flow charts.

15. References: The fast Fourier transform is discussed in:

(1) Bingham, C., M.D. Godfrey, and J.W. Tukey, "Modern Techniques of Power Spectrum Estimation." IEEE Trans. Audio Electroacoustics, Vol. AU-15, no. 2, (June 1967), pp. 56-66.

(2) Bergland, G.D., "A Fast Fourier Transform Algorithm for Real-Valued Series." Commun. ACM, Vol. 11, no.10, (Oct. 1968), pp. 703-710.

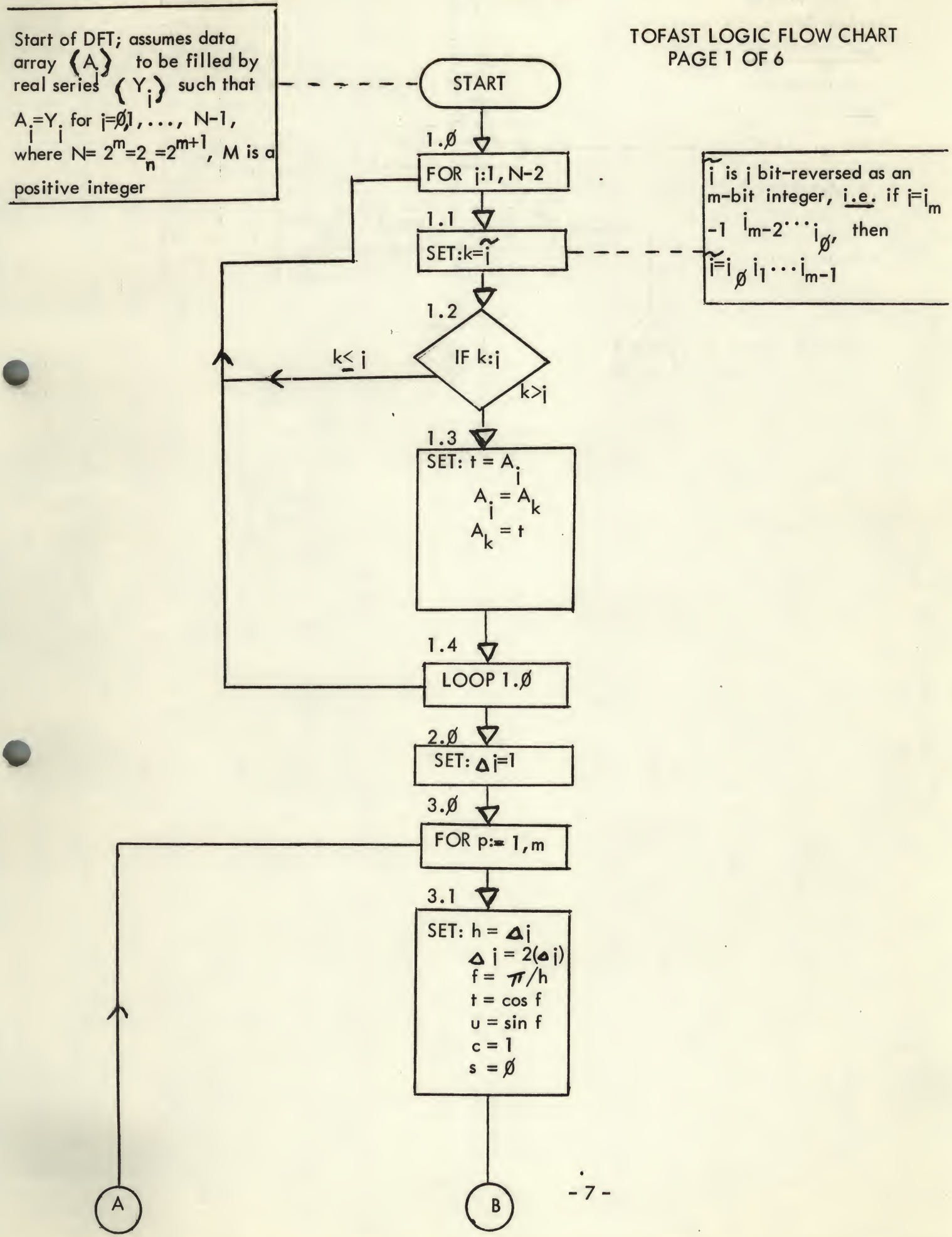
(3) Cooley, J.W. and J.W. Tukey, "An Algorithm for the Machine Calculation of Fourier Series." Math. Comput., Vol. 19, (1965), pp. 297-301.

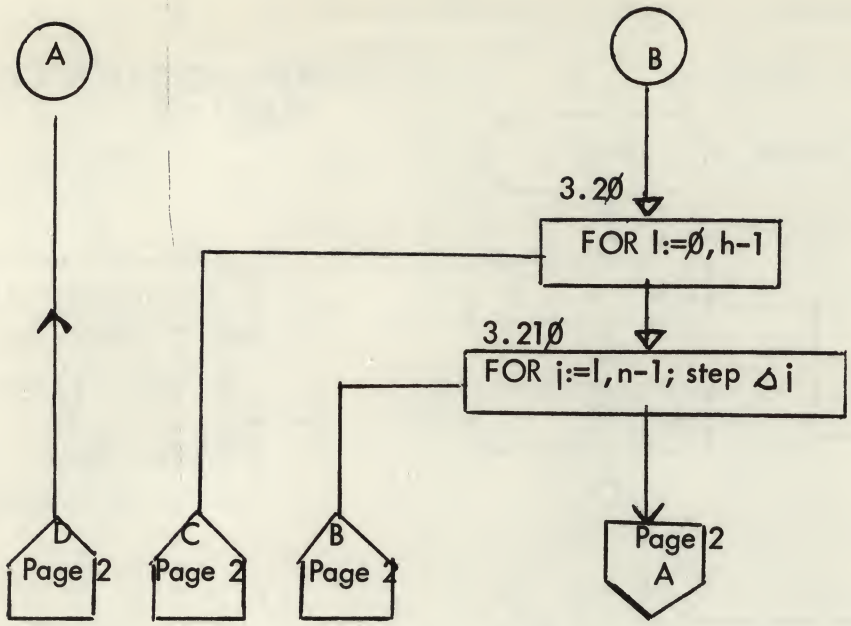
(4) Jennrich, R.I., B. Gilchrist, and A. Lynn, "REFAST: Fast Fourier Transform Routine." Program write-up of the Health Sciences Computing Facilities, UCLA, (Nov. 1967).

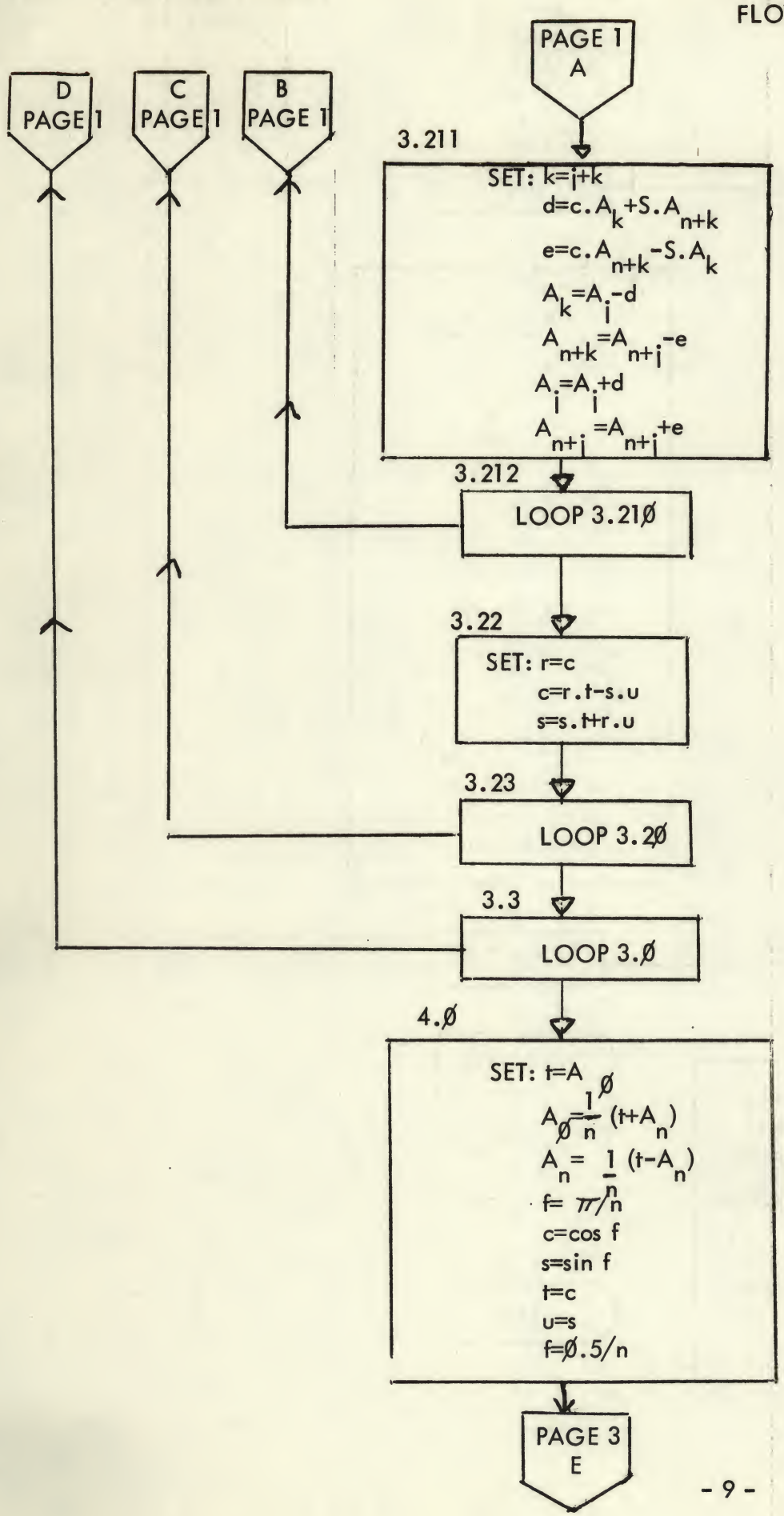
(5) Rothman, J.E., "The Fast Fourier Transform and its Implementation." Decuscope, Vol. 7, no.3, (1968), pp. 3-10.

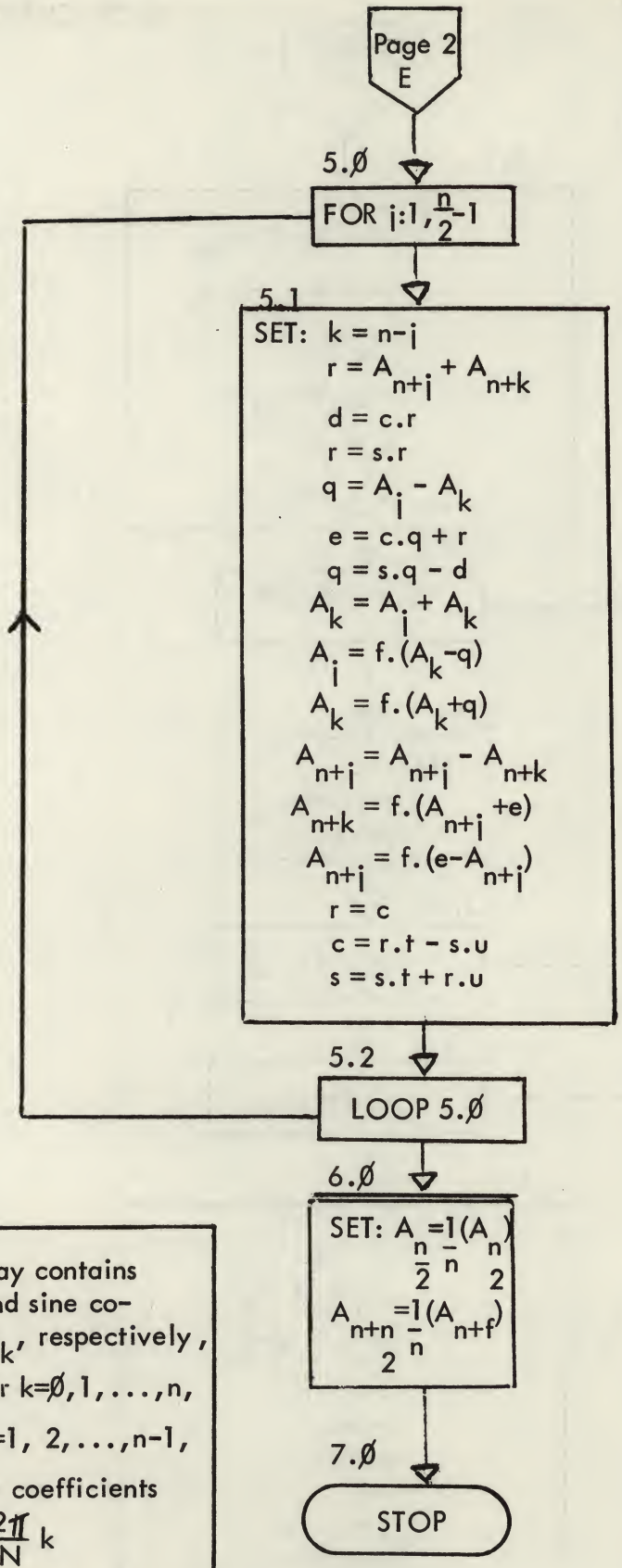
(6) Singleton, R.C., "On Computing the Fast Fourier Transform." Commun. ACM., Vol. 10, no. 10, (Oct. 1967), pp. 646-654.

(7) Walton, P.L., "On the Fast Fourier Transform," in preparation.


Start of DFT; assumes data array $\{A_i\}$ to be filled by real series $\{Y_i\}$ such that $A_i = Y_i$ for $i = 0, 1, \ldots, N-1$, where $N = 2^m = 2_n = 2^{m+1}$, M is a positive integer
START
1.0
FOR j:1, N-2
1.1
SET: k = $\tilde{j}$
$\tilde{j}$ is j bit-reversed as an m-bit integer, i.e. if $j = i_m \, -1 \; i_{m-2} \cdots i_0$, then $\tilde{j} = i_0 \, i_1 \cdots i_{m-1}$
1.2
IF k:j
k ≤ j
k > j
1.3
SET: t = A_j
$A_j = A_k$
$A_k = t$
1.4
LOOP 1.0
2.0
SET: Δj = 1
3.0
FOR p:= 1, m
3.1
SET: h = Δj
Δj = 2(Δj)
f = π/h
t = cos f
u = sin f
c = 1
s = 0
A
B


A
B
3.2Ø
FOR I:=Ø,h-1
3.21Ø
FOR j:=I,n-1; step Δj
D
Page 2
C
Page 2
B
Page 2
Page
A
2


PAGE 1
A
D
PAGE 1
C
PAGE 1
B
PAGE 1
3.211
SET: k=j+k
d=c.A_k+S.A_n+k
e=c.A_n+k-S.A_k
A_k=A_j-d
A_n+k=A_n+j-e
A_j=A_j+d
A_n+j=A_n+j+e
3.212
LOOP 3.21Ø
3.22
SET: r=c
c=r.t-s.u
s=s.t+r.u
3.23
LOOP 3.2Ø
3.3
LOOP 3.Ø
4.Ø
SET: t=A_Ø
A_Ø=1/n (t+A_n)
A_n=1/n (t-A_n)
f=π/n
c=cos f
s=sin f
t=c
u=s
f=Ø.5/n
PAGE 3
E



End of DFT; data array contains the Fourier cosine and sine coefficients, A_k and b_k, respectively, such that $a_k = A_k$ for k=Ø,1,...,n, and $b_k = A_{n+k}$ for k=1, 2,...,n-1, and where a_k, b_k are coefficients at frequency $w_k = \frac{2\pi}{N} k$

Start of DFT^{-1}; assumes data array $\{A_i\}$ to be filled by the Fourier cosine and sine coefficients, a_i and b_i respectively, such that $A_i = a_i$ for $i = \emptyset, 1, \ldots, n$, $A_{n+i} = b_i$ for $i = 1, 2, \ldots, A-1$, and where a_i, b_i are coefficients at frequency $w_i = \frac{2\pi}{N} i$,

$N = 2_n$

START

1.Ø
SET: $t = A_\emptyset$
$A_\emptyset = \frac{1}{2}(t + A_n)$
$A_n = \frac{1}{2}(t + A_n)$
$f = \pi/n$
$c = \cos f$
$s = \sin f$
$t = c$
$u = s$
$f = \emptyset.5$

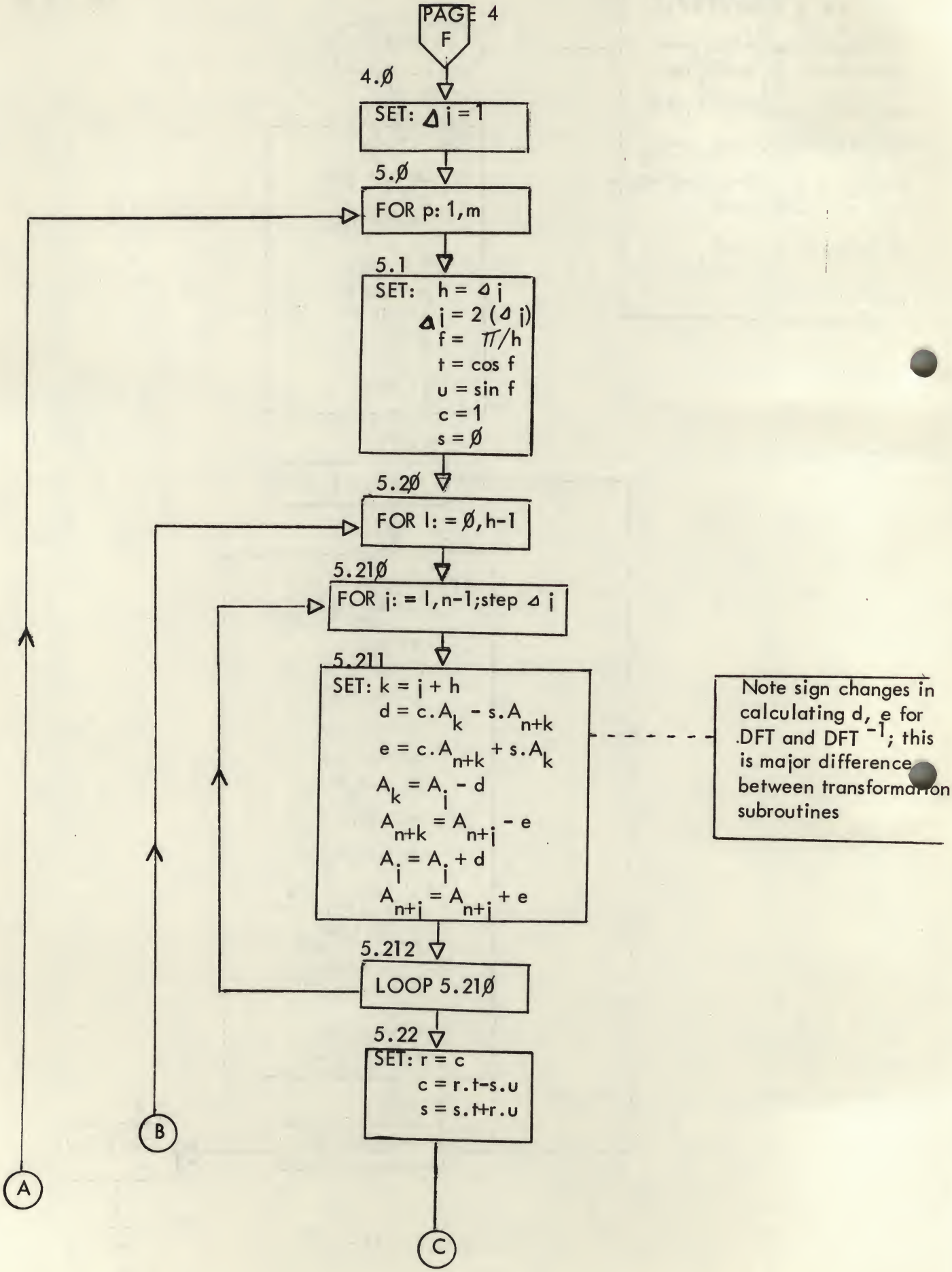
2.Ø
FOR $i := 1, \frac{n}{2} - 1$

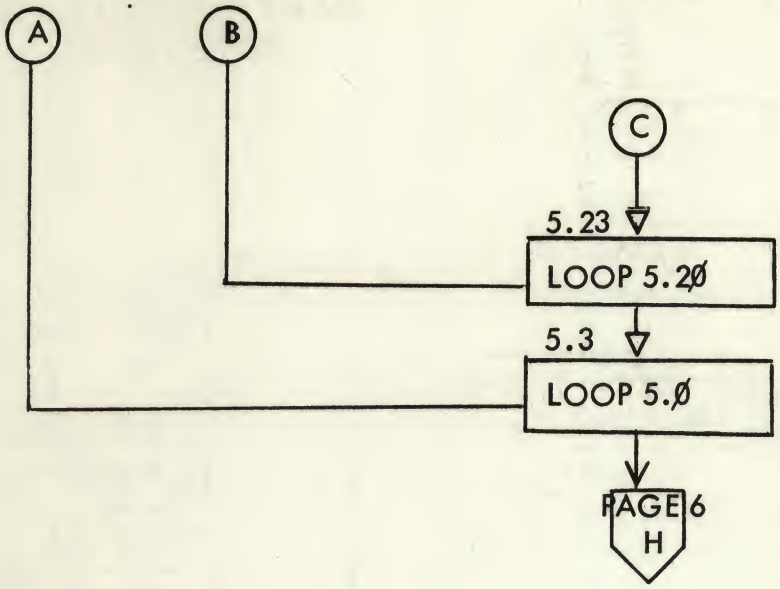
2.1
SET: $k = n - i$
$r = A_{n+i} + A_{n+k}$
$d = c.r$
$r = s.r$
$q = A_i - A_k$
$e = c.q + r$
$q = s.q - d$
$A_k = A_i + A_k$
$A_i = f.(A_k - q)$
$A_k = f.(A_k + q)$
$A_{n+i} = A_{n+i} - A_{n+k}$
$A_{n+k} = f.(A_{n+i} + e)$
$A_{n+i} = f.(e - A_{n+i})$
$r = c$
$c = r.t - s.u$
$s = s.t + r.u$

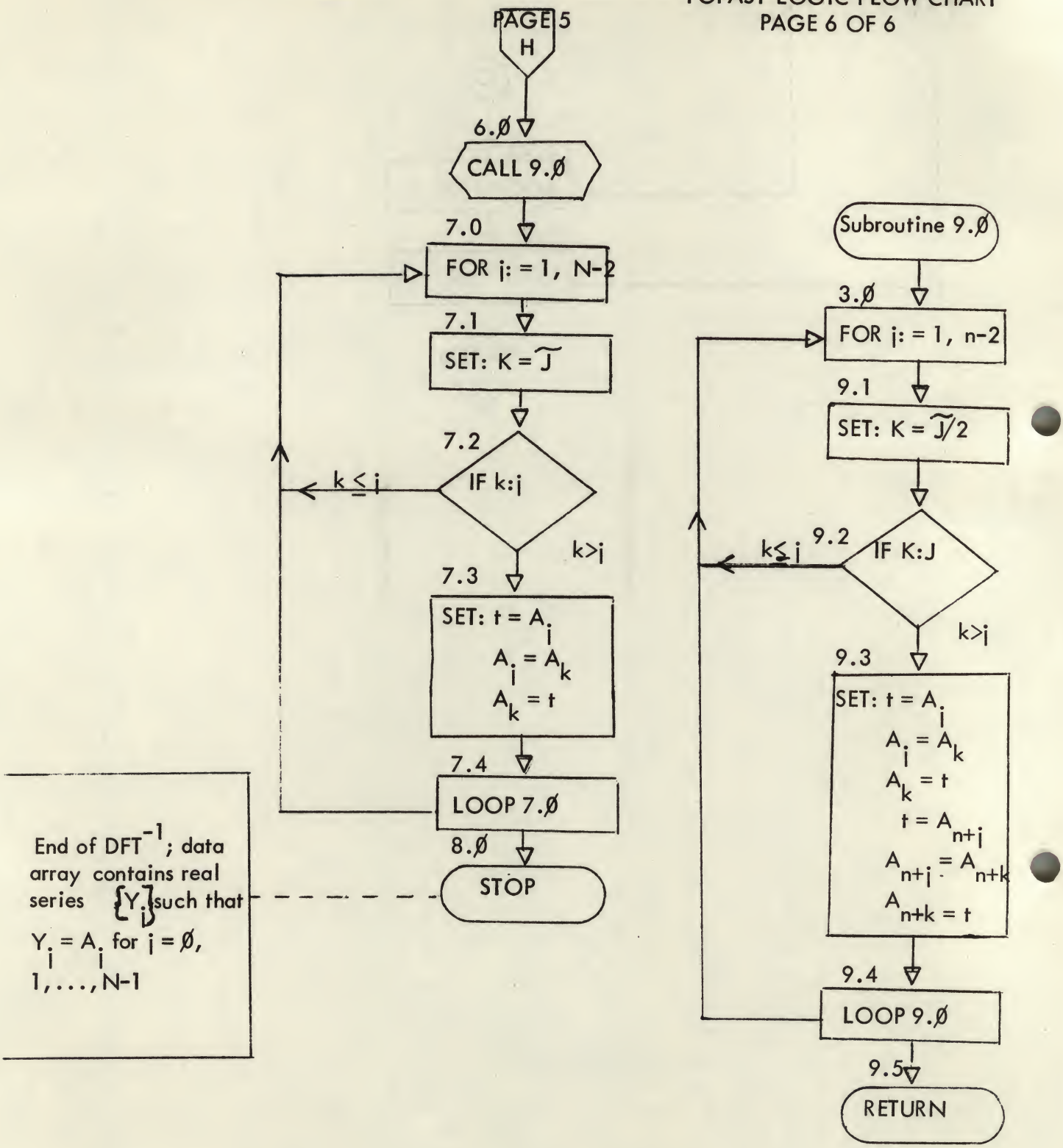
2.2
LOOP 2.Ø

3.Ø
CALL 9.Ø

Page 5
F


PAGE 4
F
4.Ø
SET: Δ j = 1
5.Ø
FOR p: 1, m
5.1
SET: h = Δ j
Δ j = 2 (Δ j)
f = π/h
t = cos f
u = sin f
c = 1
s = Ø
5.2Ø
FOR l: = Ø, h-1
5.21Ø
FOR j: = l, n-1; step Δ j
5.211
SET: k = j + h
d = c.A_k - s.A_n+k
e = c.A_n+k + s.A_k
A_k = A_j - d
A_n+k = A_n+j - e
A_j = A_j + d
A_n+j = A_n+j + e
Note sign changes in calculating d, e for DFT and DFT -1; this is major difference between transformation subroutines
5.212
LOOP 5.21Ø
5.22
SET: r = c
c = r.t-s.u
s = s.t+r.u
A
B
C


A
B
C
5.23
LOOP 5.2Ø
5.3
LOOP 5.Ø
PAGE 6
H


PAGE 5
H
6.Ø
CALL 9.Ø
7.0
FOR j: = 1, N-2
7.1
SET: K = J~
7.2
IF k:j
k ≤ j
k>j
7.3
SET: t = A_j
A_j = A_k
A_k = t
7.4
LOOP 7.Ø
8.Ø
STOP
End of DFT^-1; data array contains real series {Y_i} such that
Y_i = A_i for j = Ø, 1, . . . , N-1
Subroutine 9.Ø
3.Ø
FOR j: = 1, n-2
9.1
SET: K = J~/2
9.2
IF K:J
k≤j
k>j
9.3
SET: t = A_j
A_j = A_k
A_k = t
t = A_n+j
A_n+j = A_n+k
A_n+k = t
9.4
LOOP 9.Ø
9.5
RETURN

```
                /PROGRAM TOFAST - FAST FORWARD OR INVERSE DISCRETE
                /FOURIER TRANSFORM FOR REAL DATA
                /PAGE Ø OF TOFAST
                        *2Ø
                /TOFAST CONSTANTS
ØØ2Ø  Ø2ØØ      DATAST,  2ØØ                    /SA OF DATA ARRAY
ØØ21  ØØØ2      PI,      2
ØØ22  311Ø               311Ø
ØØ23  3755               3755
ØØ24  Ø262      INDEXL,  INDEX
                /TOFAST VARIABLES
ØØ25  ØØØØ      M,       Ø
ØØ26  ØØØØ      N,       Ø
ØØ27  ØØØØ      NMIN1,   Ø
ØØ3Ø  ØØØØ      NN,      Ø
ØØ31  ØØØØ      J,       Ø
ØØ32  ØØØØ      K,       Ø
ØØ33  ØØØØ      AJL,     Ø
ØØ34  ØØØØ      ANPJL,   Ø
ØØ35  ØØØØ      AKL,     Ø
ØØ36  ØØØØ      ANPKL,   Ø
ØØ37  ØØØØ      H,       Ø
                        *64
ØØ64  ØØØØ      C,       Ø
ØØ65  ØØØØ               Ø
ØØ66  ØØØØ               Ø
ØØ67  ØØØØ      S,       Ø
ØØ7Ø  ØØØØ               Ø
ØØ71  ØØØØ               Ø
ØØ72  ØØØØ      T,       Ø
ØØ73  ØØØØ               Ø
ØØ74  ØØØØ               Ø
ØØ75  ØØØØ      U,       Ø
ØØ76  ØØØØ               Ø
ØØ77  ØØØØ               Ø
                /PAGE 1 OF TOFAST
                        *2ØØ
                /SUBROUTINE TO DO PARTIAL TRANSFORM OF INTERMEDIATE
                /VECTOR
Ø2ØØ  ØØØØ      LOOP3,   Ø
Ø2Ø1  1Ø31               TAD J
Ø2Ø2  1Ø37               TAD H
Ø2Ø3  3Ø32               DCA K                  /K=J+H
Ø2Ø4  4242               JMS SETLOC             /GET ADDRESSES
Ø2Ø5  44Ø7               FENTER
Ø2Ø6  5436               FGET I ANPKL
Ø2Ø7  3Ø67               FMPY S
Ø21Ø  6272               FPUT D1
Ø211  5435               FGET I AKL
Ø212  3Ø64               FMPY C
```

```
Ø213  1272  SWIT1,    FADD D1                  /OR FSUB D1 'FOR INVERSE
Ø214  6272            FPUT D1                  /D1=C*A(K)+S*A(N+K)
Ø215  5435            FGET I AKL               /OR D1=C*A(K)-S*A(N+K)
Ø216  3Ø67            FMPY S
Ø217  6275            FPUT E1
Ø22Ø  5436            FGET I ANPKL
Ø221  3Ø64            FMPY C
Ø222  2275  SWIT2,    FSUB E1                  /OR FADD E1 FOR INVERSE
Ø223  6275            FPUT E1                  /E1=C*A(N+K)-S*A(K)
Ø224  5433            FGET I AJL               /OR E1=C*A(N+ K)+S*A(K)
Ø225  2272            FSUB D1
Ø226  6435            FPUT I AKL               /A(K)=A(J)-D1
Ø227  5434            FGET I ANPJL
Ø23Ø  2275            FSUB E1
Ø231  6436            FPUT I ANPKL             /A(N+K)=A(N+J)-E1
Ø232  5433            FGET I AJL
Ø233  1272            FADD D1
Ø234  6433            FPUT I AJL               /A(J)=A(J)+D1
Ø235  5434            FGET I ANPJL
Ø236  1275            FADD E1
Ø237  6434            FPUT I ANPJL             /A(N+J)=A(N+J)+E1
Ø24Ø  ØØØØ            FEXT
Ø241  56ØØ            JMP I LOOP3              /EXIT
                /SUBROUTINE TO FIND FLOATING ADDRESSES OF A(J),A(N+J),
                /A(K), AND A(N+K)
Ø242  ØØØØ  SETLOC,   Ø
Ø243  1Ø31            TAD J
Ø244  4262            JMS INDEX
Ø245  3Ø33            DCA AJL                  /ADDRESS OF A(J)
Ø246  1Ø26            TAD N
Ø247  1Ø31            TAD J
Ø25Ø  4262            JMS INDEX
Ø251  3Ø34            DCA ANPJL                /ADDRESS OF A(N+J)
Ø252  1Ø32            TAD K
Ø253  4262            JMS INDEX
Ø254  3Ø35            DCA AKL                  /ADDRESS OF A(K)
Ø255  1Ø26            TAD N
Ø256  1Ø32            TAD K
Ø257  4262            JMS INDEX
Ø26Ø  3Ø36            DCA ANPKL                /ADDRESS OF A(N+K)
Ø261  5642            JMP I SETLOC             /EXIT
                /SUBROUTINE TO FIND FLOATING ADDRESS OF A[C(AC)]
Ø262  ØØØØ  INDEX,    Ø
Ø263  3271            DCA INXSTR
Ø264  1271            TAD INXSTR
Ø265  71Ø4            CLL RAL
Ø266  1271            TAD INXSTR
Ø267  1Ø2Ø            TAD DATAST
Ø27Ø  5662            JMP I INDEX
```

```
0271 0000   INXSTR,     0
0272 0000   D1,         0
0273 0000               0
0274 0000               0
0275 0000   E1,         0
0276 0000               0
0277 0000               0
            /PAGE 2 OF TOFAST
                        *400
            /CONTROL ROUTINE; ENTER AT 0400 FOR DFT;
            /ENTER AT 0401 FOR INVERSE DFT
0400 5202               JMP FWDFFT
0401 5324               JMP REVFFT
            /DFT; ACCEPTS 2↑MP1 FIXED- OR FLOATING-POINT DATA,
            /WHERE MP1=3,4,5,...,10
0402 4257   FWDFFT,     JMS PRELIM          /SET UP CONSTANTS, FIELD
0403 7402               HLT                 /HALT FOR DATA FORMAT
0404 7604               LAS                 /BIT 0 OF SR SET IF
0405 7510               SPA                 /FLOATING;OTHERWISE SET
0406 5221               JMP DATFLT          /TO #BITS RIGHT OF .
0407 7041               CIA                 /FIXED-POINT DATA
0410 1215               TAD C13
0411 3757               DCA I BINPNL        /#BITS LEFT OF .
0412 4616               JMS I FIXPML        /PERMUTE TO COMPLEX
0413 4617               JMS I SPREAL        /VECTOR,THEN FLOAT
0414 5223               JMP DODFT
0415 0013   C13,        13
0416 0600   FIXPML,     FIXPMT
0417 1121   SPREAL,     SPREAD
0420 0606   FLTPML,     FLTPMT
0421 7300   DATFLT,     CLA CLL             /FLOATING-POINT DATA
0422 4620               JMS I FLTPML        /PERMUTE TO COMPLEX
0423 1245   DODFT,      TAD FADDD1
0424 3647               DCA I SWIT1L
0425 1246               TAD FSUBE1
0426 3650               DCA I SWIT2L        /SET LOGIC FOR DFT
0427 4654               JMS I MAINL         /DO DFT
0430 1255               TAD FDIVFN
0431 3651               DCA I SWIT3L
0432 1255               TAD FDIVFN
0433 3652               DCA I SWIT4L
0434 1255               TAD FDIVFN
0435 3653               DCA I SWIT5L        /SET LOGIC FOR UNSCRAM.
0436 4656               JMS I UNSCRL        /UNSCRAMBLE
0437 7000               NOP                 /JUMP FROM HERE TO USER-
0440 4407               FENTER              /DEFINED ROUTINES
0441 0017               FCDF 0              /SET FLOATING DATA
0442 0000               FEXT                /FIELD=INST. FIELD
0443 7000               NOP                 /JUMP FROM HERE,TOO
0444 7402               HLT                 /HALT
```

```
Ø445  1272  FADDD1,   1272                    /FADD D1 ON PAGE 1
Ø446  2275  FSUBE1,   2275                    /FSUB E1 ON PAGE 1
Ø447  Ø213  SWIT1L,   SWIT1
Ø45Ø  Ø222  SWIT2L,   SWIT2
Ø451  1244  SWIT3L,   SWIT3
Ø452  125Ø  SWIT4L,   SWIT4
Ø453  1265  SWIT5L,   SWIT5
Ø454  1ØØØ  MAINL,    MAIN
Ø455  4361  FDIVFN,   4361                    /FDIV FN ON PAGE 5
Ø456  12ØØ  UNSCRL,   UNSCRM
            /SUBROUTINE TO SET UP CONSTANTS FOR TRANSFORM,
            /AND SET FLOATING DATA FIELD
Ø457  ØØØØ  PRELIM,   Ø
Ø46Ø  76Ø4  LAS                               /ENTERING SR CONTAINS
Ø461  3Ø3Ø            DCA NN                  /NN=NO. OF REAL DATA
Ø462  1Ø3Ø            TAD NN
Ø463  1322            TAD M1Ø
Ø464  751Ø            SPA
Ø465  5271            JMP ERRNN
Ø466  1323            TAD M1771
Ø467  771Ø            SPA CLA                 /IS 7<NN<1Ø25
Ø47Ø  5274            JMP .+4
Ø471  724Ø  ERRNN,    STA                     /NO-HALT WITH 7777 IN
Ø472  74Ø2            HLT                     /AC; RESET SR TO LEGAL
Ø473  526Ø            JMP PRELIM+1            /NN, PRESS CONTINUE
Ø474  376Ø            DCA I MP1L              /YES
Ø475  1Ø3Ø            TAD NN
Ø476  711Ø            CLL RAR
Ø477  743Ø            SZL
Ø5ØØ  53Ø3            JMP .+3
Ø5Ø1  276Ø            ISZ I MP1L
Ø5Ø2  5276            JMP .-4
Ø5Ø3  764Ø            SZA CLA                 /IS NN OF FORM 2↑MP1
Ø5Ø4  5271            JMP ERRNN               /NO-ERROR HALT
Ø5Ø5  724Ø            STA                     /YES
Ø5Ø6  176Ø            TAD I MP1L
Ø5Ø7  3Ø25            DCA M                   /M=MP1-1
Ø51Ø  1Ø3Ø            TAD NN
Ø511  711Ø            CLL RAR
Ø512  3Ø26            DCA N                   /N=NN/2
Ø513  724Ø            STA
Ø514  1Ø26            TAD N
Ø515  3Ø27            DCA NMIN1               /N-1
Ø516  44Ø7            FENTER
Ø517  Ø117            FCDF 1ØØ                /SET FOR FLOATING DATA
Ø52Ø  ØØØØ            FEXT                    /FIELD=1
Ø521  5657            JMP I PRELIM            /EXIT
```

```
Ø522 777Ø  M1Ø,      -1Ø
Ø523 6ØØ7  M1771,    -1771
           /INVERSE DFT; ACCEPTS 2↑MP1 FLOATING-POINT FOURIER
           /COEFFICIENTS, WHERE MP1=3,4,5,...,1Ø, AND COEFF.S
           /ARE IN SEQUENCE A(Ø),A(1),...,A(N),B(1),B(2),...,B(N-1)
Ø524 4257  REVFFT,   JMS PRELIM          /SET UP CONSTANTS,FIELD
Ø525 1352            TAD FMPYP5
Ø526 3651            DCA I SWIT3L
Ø527 1352            TAD FMPYP5
Ø53Ø 3652            DCA I SWIT4L
Ø531 1353            TAD LFNOP
Ø532 3653            DCA I SWIT5L        /SET LOGIC FOR SCRAMBLE
Ø533 4656            JMS I UNSCRL        /SCRAMBLE
Ø534 4754            JMS I CPXPML        /PERMUTE COMPLEX VECTOR
Ø535 1355            TAD FSUBD1
Ø536 3647            DCA I SWIT1L
Ø537 1356            TAD FADDE1
Ø54Ø 365Ø            DCA I SWIT2L        /SET LOGIC FOR INVERSE
Ø541 4654            JMS I MAINL         /DO INVERSE DFT
Ø542 4754            JMS I CPXPML
Ø543 462Ø            JMS I FLTPML        /PUT DATA IN- SEQUENCE
Ø544 7ØØØ            NOP                 /JUMP FROM HERE TO USER-
Ø545 44Ø7            FENTER              /DEFINED ROUTINES
Ø546 ØØ17            FCDF Ø              /SET FLOATING DATA
Ø547 ØØØØ            FEXT                /FIELD=INST. FIELD
Ø55Ø 7ØØØ            NOP                 /OR FROM HERE
Ø551 74Ø2            HLT                 /HALT
Ø552 3351  FMPYP5,   3351                /FMPY ZP5 ON PAGE 5
Ø553 ØØ1Ø  LFNOP,    FNOP
Ø554 Ø625  CPXPML,   CPXPMT
Ø555 2272  FSUBD1,   2272                /FSUB D1 ON PAGE 1
Ø556 1275  FADDE1,   1275                /FADD E1 ON PAGE 1
Ø557 1176  BINPNL,   BINPNT
Ø56Ø Ø765  MP1L,     MP1
           /PAGE 3 OF TOFAST
                     *6ØØ
           /SUBROUTINE TO BIT-REVERSE PERMUTE NN FIXED-POINT DATA
Ø6ØØ ØØØØ  FIXPMT,   Ø
Ø6Ø1 12Ø5            TAD JMSFXP
Ø6Ø2 3253            DCA PMTMOV          /SET LOGIC
Ø6Ø3 4214            JMS PMTDAT          /PERMUTE
Ø6Ø4 56ØØ            JMP I FIXPMT        /EXIT
Ø6Ø5 43Ø1  JMSFXP,   JMS FIXMOV
           /SUBROUTINE TO BIT-REVERSE PERMUTE NN FLOATING DATA
Ø6Ø6 ØØØØ  FLTPMT,   Ø
Ø6Ø7 1213            TAD JMSFLT
Ø61Ø 3253            DCA PMTMOV          /SET LOGIC
Ø611 4214            JMS PMTDAT          /PERMUTE
Ø612 56Ø6            JMP I FLTPMT        /EXIT
```

```
0613  4321  JMSFLT,   JMS FLTMOV
            /SUBROUTINE TO SET UP FOR PERMUTION OF NN REAL DATA
0614  0000  PMTDAT,   0
0615  1224            TAD LNOP
0616  3275            DCA FLPCTL              /SET LOGIC
0617  1027            TAD NMIN1
0620  7104            CLL RAL
0621  7041            CIA                     /NN-2 DATA
0622  4240            JMS PERMUT              /PERMUTE
0623  5614            JMP I PMTDAT            /EXIT
0624  7000  LNOP,     NOP
            /SUBROUTINE TO BIT-REVERSE PERMUTE N FLOATING-POINT
            /COMPLEX DATA
0625  0000  CPXPMT,   0
0626  1237            TAD JMSCPX
0627  3253            DCA PMTMOV
0630  1267            TAD CLLRAR
0631  3275            DCA FLPCTL              /SET LOGIC
0632  7240            STA
0633  1027            TAD NMIN1
0634  7041            CIA                     /N-2 DATA
0635  4240            JMS PERMUT              /PERMUTE
0636  5625            JMP I CPXPMT            /EXIT
0637  4341  JMSCPX,   JMS CPXMOV
            /SUBROUTINE TO BIT-REVERSE PERMUTE -C(AC) DATA AS
            /SET UP BY LOGIC ALTERATIONS
0640  0000  PERMUT,   0
0641  3360            DCA CTR1                /-NO. OF DATA
0642  7201            CLA IAC
0643  3031            DCA J                   /START WITH J=1
0644  1031  PMTNXT,   TAD J
0645  4260            JMS FLIP
0646  3032            DCA K                   /K=BIT-REVERSED J
0647  1032            TAD K
0650  7041            CIA
0651  1031            TAD J
0652  7710            SPA CLA                 /IS K>J
0653  0000  PMTMOV,   0                       /YES-SWITCH DATA J,K
0654  2031            ISZ J                   /NEXT J
0655  2360            ISZ CTR1                /HAVE ALL DATA BEEN DONE
0656  5244            JMP PMTNXT              /NO-NEXT
0657  5640            JMP I PERMUT            /YES-EXIT
            /SUBROUTINE TO BIT REVERSE C(AC) AS MP1- OR M-BIT WORD
0660  0000  FLIP,     0
0661  3277            DCA FLPJ
0662  1365            TAD MP1
0663  7041            CIA
0664  3361            DCA CTR2                /MP1 BITS TO BE REV.
```

```
0665  3300  FLPLOP,   DCA FLPK
0666  1277            TAD FLPJ
0667  7110  CLLRAR,   CLL RAR              /READ J FROM R TO L
0670  3277            DCA FLPJ
0671  1300            TAD FLPK
0672  7004            RAL                  /BUILD K FROM L TO R
0673  2361            ISZ CTR2             /HAVE ALL BITS BEEN REV.
0674  5265            JMP FLPLOP           /NO-NEXT
0675  7000  FLPCTL,   NOP                  /YES (CLL RAR FOR M-BIT)
0676  5660            JMP I FLIP           /EXIT
0677  0000  FLPJ,     0
0700  0000  FLPK,     0
            /SUBROUTINE TO SWITCH FIXED-POINT A(J) AND A(K)
0701  0000  FIXMOV,   0
0702  1031            TAD J
0703  1020            TAD DATAST
0704  3033            DCA AJL              /ADDRESS OF A(J)
0705  1032            TAD K
0706  1020            TAD DATAST
0707  3035            DCA AKL              /ADDRESS OF A(K)
0710  6211            CDF 10
0711  1435            TAD I AKL
0712  3362            DCA T3               /T3=A(K)
0713  1433            TAD I AJL
0714  3435            DCA I AKL            /A(K)=A(J)
0715  1362            TAD T3
0716  3433            DCA I AJL            /A(J)=T3
0717  6201            CDF 0
0720  5701            JMP I FIXMOV         /EXIT
            /SUBROUTINE TO SWITCH FLOATING-POINT DATA A(J) AND A(K)
0721  0000  FLTMOV,   0
0722  1031            TAD J
0723  4424            JMS I INDEXL
0724  3033            DCA AJL              /ADDRESS OF A(J)
0725  1032            TAD K
0726  4424            JMS I INDEXL
0727  3035            DCA AKL              /ADDRESS OF A(K)
0730  4407            FENTER
0731  5435            FGET I AKL
0732  6362            FPUT T3              /T3=A(K)
0733  5433            FGET I AJL
0734  6435            FPUT I AKL           /A(K)=A(J)
0735  5362            FGET T3
0736  6433            FPUT I AJL           /A(J)=T3
0737  0000            FEXT
0740  5721            JMP I FLTMOV         /EXIT
            /SUBROUTINE TO SWITCH COMPLEX DATA S(J) AND S(K)
```

```
ø741  ØØØØ   CPXMOV,   ø
ø742  4321             JMS FLTMOV          /SWITCH REAL PARTS
ø743  1ø31             TAD J
ø744  3357             DCA SAVEJ           /SAVE J
ø745  1ø26             TAD N
ø746  1ø31             TAD J
ø747  3ø31             DCA J               /IM OF S(J)=A(N+J)
ø75ø  1ø26             TAD N
ø751  1ø32             TAD K
ø752  3ø32             DCA K               /IM OF S(K)=A(N+K)
ø753  4321             JMS FLTMOV          /SWITCH IMAGIN. PARTS
ø754  1357             TAD SAVEJ
ø755  3ø31             DCA J               /RESTORE J
ø756  5741             JMP I CPXMOV        /EXIT
ø757  ØØØØ   SAVEJ,    ø
ø76ø  ØØØØ   CTR1,     ø
ø761  ØØØØ   CTR2,     ø
ø762  ØØØØ   T3,       ø
ø763  ØØØØ             ø
ø764  ØØØØ             ø
ø765  ØØØØ   MP1,      ø
             /PAGE 4 OF TOFAST
                       *1ØØØ
             /SUBROUTINE TO TRANSFORM N FLOATING-POINT COMPLEX DATA
1ØØØ  ØØØØ   MAIN,     ø
1ØØ1  72ø1             CLA IAC
1ØØ2  3367             DCA DJ              /START WITH DJ=1
1ØØ3  1ø25             TAD M
1ØØ4  7ø41             CIA
1ØØ5  337ø             DCA MINP            /M INTERMEDIATE TRANS.
1ØØ6  4236   PLOOP,    JMS LOOP1           /SET UP FOR INT. TRANS.
1ØØ7  3371             DCA L               /L=ø
1ø1ø  1ø37             TAD H
1ø11  7ø41             CIA
1ø12  3372             DCA MINL            /H PART./INT.
1ø13  1371   LLOOP,    TAD L
1ø14  3ø31             DCA J               /J=L
1ø15  4635   JLOOP,    JMS I LOOP3L        /DO INT. PARTIAL
1ø16  1ø31             TAD J
1ø17  1367             TAD DJ
1ø2ø  3ø31             DCA J               /J=J+DJ
1ø21  1ø31             TAD J
1ø22  7ø41             CIA
1ø23  1ø27             TAD NMIN1
1ø24  77ØØ             SMA CLA             /IS J>N-1
1ø25  5215             JMP JLOOP           /NO-NEXT PASS OF PART.
1ø26  2372             ISZ MINL            /YES-ARE ALL PART.S DONE
1ø27   5233            JMP MOREL           /NO-NEXT PARTIAL
```

```
1Ø3Ø  237Ø          ISZ MINP          /YES-ARE ALL INT.S DONE
1Ø31  52Ø6          JMP PLOOP         /NO-NEXT INTERMEDIATE
1Ø32  56ØØ          JMP I MAIN        /YES-EXIT
1Ø33  4273  MOREL,  JMS SETUPL        /SET UP FOR NEXT PARTIAL
1Ø34  5213          JMP LLOOP
1Ø35  Ø2ØØ  LOOP3L, LOOP3
          /SUBROUTINE TO SET UP FOR INTERMEDIATE TRANSFORM
1Ø36  ØØØØ  LOOP1,  Ø
1Ø37  1367          TAD DJ
1Ø4Ø  3Ø37          DCA H             /H=DJ
1Ø41  1367          TAD DJ
1Ø42  71Ø4          CLL RAL
1Ø43  3367          DCA DJ            /DJ=2*DJ
1Ø44  1Ø37          TAD H
1Ø45  3Ø45          DCA 45
1Ø46  44Ø7          FENTER
1Ø47  ØØ11          FLOAT
1Ø5Ø  6373          FPUT R4
1Ø51  5Ø21          FGET PI
1Ø52  4373          FDIV R4
1Ø53  6373          FPUT R4           /R4=PI/H
1Ø54  ØØØ4          FCOS
1Ø55  6Ø72          FPUT T            /T=CO S(R4)
1Ø56  5373          FGET R4
1Ø57  ØØØ3          FSIN
1Ø6Ø  6Ø75          FPUT U            /U=SIN(R4)
1Ø61  ØØØØ          FEXT
1Ø62  72Ø1          CLA IAC
1Ø63  3Ø64          DCA C
1Ø64  7132          STL RTR
1Ø65  3Ø65          DCA C+1
1Ø66  3066          DCA C+2           /C=1
1Ø67  3Ø67          DCA S
1Ø7Ø  3Ø7Ø          DCA S+1
1Ø71  3Ø71          DCA S+2
1Ø72  5636          JMP I LOOP1       /EXIT
          /SUBROUTINE TO SET UP FOR INTERMEDIATE PARTIAL TRANS-
          /FORM EXCEPT FOR FIRST ONE
1Ø73  ØØØØ  SETUPL, Ø
1Ø74  44Ø7          FENTER
1Ø75  5Ø64          FGET C
1Ø76  6373          FPUT R4           /R4=C
1Ø77  5Ø67          FGET S
11ØØ  3Ø75          FMPY U
11Ø1  6Ø64          FPUT C
11Ø2  5373          FGET R4
11Ø3  3Ø72          FMPY T
```

```
1104 2064              FSUB C
1105 6064              FPUT C                    /C=R4*T-S*U
1106 5067              FGET S
1107 3072              FMPY T
1110 6067              FPUT S
1111 5373              FGET P4
1112 3075              FMPY U
1113 1067              FADD S
1114 6067              FPUT S                    /S=S*T+R4*U
1115 0000              FEXT
1116 2371              ISZ L                     /L=L+1
1117 7000              NOP
1120 5673              JMP I SETUPL              /EXIT
                /SUBROUTINE TO FLOAT NN FIXED-POINT DATA
1121 0000   SPREAD,    0
1122 1030              TAD NN
1123 7041              CIA
1124 3377              DCA MINJ4                 /NN DATA
1125 7240              STA
1126 1030              TAD NN
1127 3031              DCA J                     /START AT END
1130 1031              TAD J
1131 4424              JMS I INDEXL
1132 3365              DCA SPDFLT                /ADD. OF LAST FLOAT
1133 1031              TAD J
1134 1020              TAD DATAST
1135 3366              DCA SPDFIX                /ADD. OF LAST FIXED
1136 6211   SPDNXT,    CDF 10
1137 1766              TAD I SPDFIX
1140 6201              CDF 0
1141 3045              DCA 45
1142 3046              DCA 46
1143 3047              DCA 47
1144 1376              TAD BINPNT
1145 3044              DCA 44
1146 4764              JMS I DNORML              /FLOAT DATA
1147 4407              FENTER
1150 6765              FPUT I SPDFLT             /PUT IT AWAY
1151 0000              FEXT
1152 7240              STA
1153 1366              TAD SPDFIX
1154 3366              DCA SPDFIX                /RETREAT POINTERS
1155 7126              STL RTL
1156 7040              CMA
1157 1365              TAD SPDFLT
1160 3365              DCA SPDFLT
1161 2377              ISZ MINJ4                 /ARE ALL FLOATING
1162 5336              JMP SPDNXT                /NO-NEXT
1163 5721              JMP I SPREAD              /YES-SWIM BACK
```

```
1164  66ØØ  DNORML,   DNORM
1165  ØØØØ  SPDFLT,   Ø
1166  ØØØØ  SPDFIX,   Ø
1167  ØØØØ  DJ,       Ø
117Ø  ØØØØ  MINP,     Ø
1171  ØØØØ  L,        Ø
1172  ØØØØ  MINL,     Ø
1173  ØØØØ  R4,       Ø
1174  ØØØØ            Ø
1175  ØØØØ            Ø
1176  ØØØØ  BINPNT,   Ø
1177  ØØØØ  MINJ4,    Ø
            /PAGE 5 OF TOFAST
                      *12ØØ
            /SUBROUTINE TO UN- OR RE-SCRAMBLE COEFFICIENTS
12ØØ  ØØØØ  UNSCRM,   Ø
12Ø1  423Ø            JMS SETUPF          /DO A(Ø),A(N);SET UP REST
12Ø2  1Ø26            TAD N
12Ø3  711Ø            CLL RAR
12Ø4  7Ø41            CIA
12Ø5  7ØØ1            IAC
12Ø6  3Ø37            DCA H               /N/2-1 ITERATIONS
12Ø7  72Ø1            CLA IAC
121Ø  3Ø31            DCA J               /START WITH J=1
1211  4271  UNLOOP,   JMS LOOP4           /UN- OR RE-SCRAMBLE
1212  2Ø31            ISZ J               /NEXT J
1213  2Ø37            ISZ H               /IS SCRAMBLING DONE
1214  5211            JMP UNLOOP          /NO-NEXT
1215  2361            ISZ FN              /YES-DO A(N/2),A(N+N/2)
1216  4755            JMS I SETLOL
1217  44Ø7            FENTER
122Ø  5433            FGET I AJL
1221  3361            FMPY FN
1222  6433            FPUT I AJL
1223  5434            FGET I ANPJL
1224  3361            FMPY FN
1225  6434            FPUT I ANPJL
1226  ØØØØ            FEXT
1227  56ØØ            JMP I UNSCRM        /EXIT
            /SUBROUTINE TO UN- OR RE-SCRAMBLE A(Ø),A(N), AND
            /SET UP FOR REST
123Ø  ØØØØ  SETUPF,   Ø
1231  1Ø26            TAD N
1232  4424            JMS I INDEXL
1233  3Ø35            DCA AKL             /ADDRESS OF A(N)
1234  1Ø26            TAD N
1235  3Ø45            DCA 45
1236  44Ø7            FENTER
```

```
1237 ØØ11              FLOAT
124Ø 6361              FPUT FN
1241 542Ø              FGET I DATAST
1242 6Ø72              FPUT T                  /T=A(Ø)
1243 1435              FADD I AKL
1244 4361  SWIT3,      FDIV FN                 /OR FMPY ZP5 FOR INVERSE
1245 642Ø              FPUT I DATAST           /A(Ø)=[A(Ø)+A(N)]/N,
1246 5Ø72              FGET T                  /OR A(Ø)=[A(Ø)+A(N)]/2
1247 2435              FSUB I AKL
125Ø 4361  SWIT4,      FDIV FN                 /OR FMPY ZP5 FOR INVERSE
1251 6435              FPUT I AKL              /A(N)= [ T-A(N) ]/N,
1252 5021              FGET PI                 /OR A(N)= [ T-A(N) ]/2
1253 4361              FDIV FN
1254 6372              FPUT F5                 /F5=PI/N
1255 ØØØ4              FCOS
1256 6Ø72              FPUT T                  /T=CO S(F5)
1257 6Ø64              FPUT C                  /C=T
126Ø 5372              FGET F5
1261 ØØØ3              FSIN
1262 6Ø75              FPUT U                  /U=SIN(F5)
1263 6Ø67              FPUT S                  /S=U
1264 5351              FGET ZP5
1265 4361  SWIT5,      FDIV FN                 /OR FNOP FOR INVERSE
1266 6361              FPUT FN                 /FN=.5/N, OR FN=.5
1267 ØØØØ              FEXT
127Ø 563Ø              JMP I SETUPF            /EXIT
           /SUBROUTINE TO DO ACTUAL UN- OR RE-SCRAMBLING
1271 ØØØØ  LOOP4,      Ø
1272 1Ø31              TAD J
1273 7Ø41              CIA
1274 1Ø26              TAD N
1275 3Ø32              DCA K                   /K=N-J
1276 4755              JMS I SETLOL            /GET ADDRESSES
1277 44Ø7              FENTER
13ØØ 5434              FGET I ANPJL
13Ø1 1436              FADD I ANPKL
13Ø2 6375              FPUT R5                 /R5=A(N+J)+A(N+K)
13Ø3 3Ø64              FMPY C
13Ø4 6364              FPUT D5                 /D5=C*R5
13Ø5 5375              FGET R5
13Ø6 3Ø67              FMPY S
13Ø7 6375              FPUT R5                 /R5=S*R5
131Ø 5433              FGET I AJL
1311 2435              FSUB I AKL
1312 6356              FPUT Q                  /Q=A(J)-A(K)
1313 3Ø64              FMPY C
1314 1375              FADD R5
1315 6367              FPUT E5                 /E5=C*Q+R5
```

```
1316 5356          FGET Q
1317 3Ø67          FMPY S
132Ø 2364          FSUB D5
1321 6356          FPUT Q                  /Q=S*Q-D5
1322 5433          FGET I AJL
1323 1435          FADD I AKL
1324 6435          FPUT I AKL              /A(K)=A(J)+A(K)
1325 2356          FSUB Q
1326 3361          FMPY FN
1327 6433          FPUT I AJL              /A(J)=[A(K)-Q]*FN
133Ø 5435          FGET I AKL
1331 1356          FADD Q
1332 3361          FMPY FN
1333 6435          FPUT I AKL              /A(K)=[A(K)+Q]*FN
1334 5434          FGET I ANPJL
1335 2436          FSUB I ANPKL
1336 6434          FPUT I ANPJL            /A(N+J)=A(N+J)-A(N+K)
1337 1367          FADD E5
134Ø 3361          FMPY FN
1341 6436          FPUT I ANPKL            /A(N+K)=[A(N+J)+E5]*FN
1342 5367          FGET E5
1343 2434          FSUB I ANPJL
1344 3361          FMPY FN
1345 6434          FPUT I ANPJL            /A(N+J)=[E5-A(N+J)]*FN
1346 ØØØØ          FEXT
1347 4754          JMS I SETULL            /SET UP FOR NEXT GROUP
135Ø 5671          JMP I LOOP4             /EXIT
1351 ØØØØ  ZP5,    Ø
1352 2ØØØ          2ØØØ
1353 ØØØØ          Ø
1354 1Ø73  SETULL, SETUPL
1355 Ø242  SETLOL, SETLOC
1356 ØØØØ  Q,      Ø
1357 ØØØØ          Ø
136Ø ØØØØ          Ø
1361 ØØØØ  FN,     Ø
1362 ØØØØ          Ø
1363 ØØØØ          Ø
1364 ØØØØ  D5,     Ø
1365 ØØØØ          Ø
1366 ØØØØ          Ø
1367 ØØØØ  E5,     Ø
137Ø ØØØØ          Ø
1371 ØØØØ          Ø
1372 ØØØØ  F5,     Ø
1373 ØØØØ          Ø
1374 ØØØØ          Ø
```

```
1375 ØØØØ   R5,        Ø
1376 ØØØØ              Ø
1377 ØØØØ              Ø
                /DEFINITIONS FOR TOFAST
                       FENTER=JMS I 7
                       DNORM=66ØØ
                $
```

AJL	ØØ33	FMPYP5	Ø552	PLOOP	1ØØ6
AKL	ØØ35	FN	1361	PMTDAT	Ø614
ANPJL	ØØ34	FSUBD1	Ø555	PMTMOV	Ø653
ANPKL	ØØ36	FSUBE1	Ø446	PMTNXT	Ø644
BINPNL	Ø557	FWDFFT	Ø4Ø2	PRELIM	Ø457
BINPNT	1176	F5	1372	Q	1356
C	ØØ64	H	ØØ37	REVFFT	Ø524
CLLRAR	Ø667	INDEX	Ø262	R4	1173
CPXMOV	Ø741	INDEXL	ØØ24	R5	1375
CPXPML	Ø554	INXSTR	Ø271	S	ØØ67
CPXPMT	Ø625	J	ØØ31	SAVEJ	Ø757
CTR1	Ø76Ø	JLOOP	1Ø15	SETLOC	Ø242
CTR2	Ø761	JMSCPX	Ø637	SETLOL	1355
C13	Ø415	JMSFLT	Ø613	SETULL	1354
DATAST	ØØ2Ø	JMSFXP	Ø6Ø5	SETUPF	123Ø
DATFLT	Ø421	K	ØØ32	SETUPL	1Ø73
DJ	1167	L	1171	SPDFIX	1166
DNORM	66ØØ	LFNOP	Ø553	SPDFLT	1165
DNORML	1164	LLOOP	1Ø13	SPDNXT	1136
DODFT	Ø423	LNOP	Ø624	SPREAD	1121
D1	Ø272	LOOP1	1Ø36	SPREAL	Ø417
D5	1364	LOOP3	Ø2ØØ	SWIT1	Ø213
ERRNN	Ø471	LOOP3L	1Ø35	SWIT1L	Ø447
E1	Ø275	LOOP4	1271	SWIT2	Ø222
E5	1367	M	ØØ25	SWIT2L	Ø45Ø
FADDD1	Ø445	MAIN	1ØØØ	SWIT3	1244
FADDE1	Ø556	MAINL	Ø454	SWIT3L	Ø451
FDIVFN	Ø455	MINJ4	1177	SWIT4	125Ø
FENTER	44Ø7	MINL	1172	SWIT4L	Ø452
FIXMOV	Ø7Ø1	MINP	117Ø	SWIT5	1265
FIXPML	Ø416	MOREL	1Ø33	SWIT5L	Ø453
FIXPMT	Ø6ØØ	MP1	Ø765	T	ØØ72
FLIP	Ø66Ø	MP1L	Ø56Ø	T3	Ø762
FLPCTL	Ø675	M1Ø	Ø522	U	ØØ75
FLPJ	Ø677	M1771	Ø523	UNLOOP	1211
FLPK	Ø7ØØ	N	ØØ26	UNSCRL	Ø456
FLPLOP	Ø665	NMIN1	ØØ27	UNSCRM	12ØØ
FLTMOV	Ø721	NN	ØØ3Ø	ZP5	1351
FLTPML	Ø42Ø	PERMUT	Ø64Ø		
FLTPMT	Ø6Ø6	PI	ØØ21		