ELECTRONOTES

APPLICATION NOTE NO. 312

1 Pheasant Lane
Ithaca, NY 14850
(607)-273-8030

December 1990

AN INTUITIVE APPROACH TO FFT ALGORITHMS

Most readers are probably familiar with the general notion of an FFT (Fast Fourier Transform). An FFT is actually any algorithm that computes the DFT (Discrete Fourier Transform) with greater efficiency than the direct calculation of the DFT. Although many people use FFT's, most such persons may only have an elementary idea as to how FFT's work, and fewer still have ever needed to actually write one.

The purpose of this note is to show a largely intuitive method of obtaining FFT flow graphs (the so-called "butterflies"), which could then be programed. The literature abounds with numerous FFT methods, with many different names, different matrix factorizations, different index mappings, different decimations, etc. In light of all these mathematical considerations, it is interesting that one can still come up with correct and usable FFT butterflies largely through intuition and experience. One starts with some familiarity with the basic structures (i.e., component DFT's), with typical interconnection schemes, and then follows this with intuitive reasoning. In addition to obtaining possible butterflies, study of this sort of flow graph by developmental thinking can be useful in aiding our understanding of how and why the FFT works.

As an example, we will consider obtaining a butterfly for a 12-point FFT. Note that FFT's for powers of 2 are by far the most common, so we are purposely taking a different example. We will show the decomposition of this 12-point DFT into four 3-point DFT's and three 4-point DFT's, and will obtain the input and output orderings.

The DFT equation is:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j(2\pi/N)nk}$$

for k = 0 to N-1. [Note that N will be 12 in our example.] That is, we need to sum up samples weighted by the so-called "phase factors" $e^{-j(2\pi/N)nk}$. The DFT can be put in the form of a matrix equation:

$$\overline{X}(k) = \mathbf{W} \; \overline{x}(n)$$

where $\overline{X}(k)$ is a length N vector usually corresponding to samples in frequency while $\overline{x}(n)$ is a length N vector usually corresponding to samples in time. The matrix **W** is a matrix of phase factors:

$$
\begin{array}{lllll}
1 & 1 & 1 & \cdots & 1 \\
1 & e^{-j(2\pi/N)} & e^{-j(2\pi/N)2} & \cdots & e^{-j(2\pi/N)(N-1)} \\
1 & e^{-j(2\pi/N)2} & e^{-j(2\pi/N)4} & \cdots & e^{-j(2\pi/N)2(N-1)} \\
1 & e^{-j(2\pi/N)3} & \cdots & & \\
\cdot & & & & \\
\cdot & & & & \\
1 & e^{-j(2\pi/N)(N-1)} & e^{-j(2\pi/N)2(N-1)} & \cdots & e^{-j(2\pi/N)(N-1)(N-1)}
\end{array}
$$

Clearly the writing of the common exponential factor $e^{-j(2\pi/N)}$ is tedious, and what is really of interest is the so-called "nk" values. Accordingly we can write an "nk-array" that contains only the nk factors. At this point, we can take on our specific example of N=12. The array is then:

0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11
0	2	4	6	8	10	12	14	16	18	20	22
0	3	6	9	12	15	18	21	24	27	30	33
0	4	8	12	16	20	24	28	32	36	40	44
0	5	10	15	20	25	30	35	40	45	50	55
0	6	12	18	24	30	36	42	48	54	60	66
0	7	14	21	28	35	42	49	56	63	70	77
0	8	16	24	32	40	48	56	64	72	80	88
0	9	18	27	36	45	54	63	72	81	90	99
0	10	20	30	40	50	60	70	80	90	100	110
0	11	22	33	44	55	66	77	88	99	110	121

Next we can reduce this array by considering all the entries to by replaced with the numbers above "MOD 12". That is, we subtract 12 from each entry until the remainder is less than 12. We do this because of the periodicity of the phase factors:

$$e^{-j(2\pi/N)nk} = e^{-j(2\pi/N)(mN + nk)}$$

where m is any integer. This ability to exploit the periodicity of the phase factors is one of the two major reasons that an FFT works. The array reduced MOD 12 is seen below:

0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11
0	2	4	6	8	10	0	2	4	6	8	10
0	3	6	9	0	3	6	9	0	3	6	9
0	4	8	0	4	8	0	4	8	0	4	8
0	5	10	3	8	1	6	11	4	9	2	7
0	6	0	6	0	6	0	6	0	6	0	6
0	7	2	9	4	11	6	1	8	3	10	5
0	8	4	0	8	4	0	8	4	0	8	4
0	9	6	3	0	9	6	3	0	9	6	3
0	10	8	6	4	2	0	10	8	6	4	2
0	11	10	9	8	7	6	5	4	3	2	1

At this point, we will use the idea that the 12-point DFT can be broken up into smaller sized DFT's. In this way, a desired phase factor may be obtained as the combination of two or more phase shifts that are encountered in an overall path through the smaller DFT's. This idea of multiple use of paths is the second of the two major reasons why an FFT works. In our case, we will build our 12-point FFT out of 3-point and 4-point DFT's, and we need to study these separately, and adjust them to meet our needs.

A 3-point DFT is calculated according to the equation:

$$X(k) = \sum_{n=0}^{2} x(n)e^{-j(2\pi/3)nk}$$

while a 4-point DFT is calculated as:

$$X(k) = \sum_{n=0}^{3} x(n)e^{-j(2\pi/4)nk}$$

The "nk-arrays" for these DFT's are clearly:

0	0	0		0	0	0	0
0	1	2		0	1	2	3
0	2	4		0	2	4	6
				0	3	6	9

which can be reduced MOD-3 and MOD-4 respectively to:

0	0	0		0	0	0	0
0	1	2		0	1	2	3
0	2	1		0	2	0	2
				0	3	2	1

The final adjustment we need to make to these smaller DFT's is that they are to be used inside a 12-point DFT rather than in 3- or 4-point DFT's. Clearly:

$$e^{-j(2\pi/N_1)nk} = e^{-j(2\pi/N_2)(N_2/N_1)nk}$$

Accordingly, to change from a 3-point to a 12-point DFT we need to multiply the 3-point nk values by 4, and to change from a 4-point to a 12-point DFT we need to multiply the 4-point nk values by 3. The final nk-arrays for use in a 12-point DFT then become:

0	0	0		0	0	0	0
0	4	8		0	3	6	9
0	8	4		0	6	0	6
				0	9	6	3

The corresponding flow graphs for the 3-point and 4-point FFT's, adjusted to work in a 12-point DFT, are shown in Fig. 1a and Fig. 1b respectively. Here the signal flow is always from left to right, and the numbers in the open circles represent the "path twist" in terms of their nk value for N=12. We can think of these two flow graphs as the basic blocks which we have available to build with. We will be using three 4-point DFT's and four 3-point DFT's in our scheme.

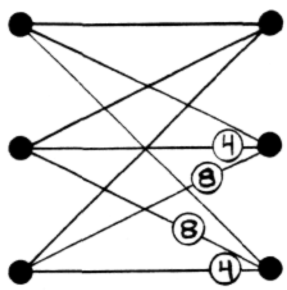

Fig. 1a
3-Point DFT

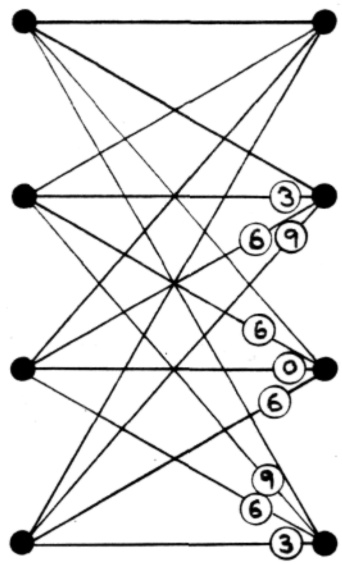

Fig. 1b
4-Point DFT

For our specific example, we will place the 7 DFT's as shown in Fig. 2, with the four 3-point DFT's on the left, and the three 4-point DFT's on the right. This figure also shows a proposed interconnection of the blocks by the 12 lines shown. We note that there are only 12 lines, that no output node of the 3-point DFT's is used more than once, and that no input node to the 4-point DFT's is used more than once.

The motivation for the particular interconnect scheme could come from a mathematical formulation. Alternatively, we can use our familiarity with similar structures and some reasoning to obtain the same result. In particular, we note that X(0) is to be obtained as the sum of all 12 inputs, with no phase shifts. This is possible by having X(0) be the top output of the top 4-point DFT, with the four inputs to this DFT being obtained as the top outputs of the four 3-point DFT's. In this way, four of the 12 interconnection paths are determined. The remaining paths can be determined by analogy with the first four, with a consideration also given to overall symmetry. That is, the middle 4-point DFT receives its inputs from the middle outputs of the four 3-point DFT's. Finally, the lower 4-point DFT receives its inputs from the lower outputs of the four 3-point DFT's.

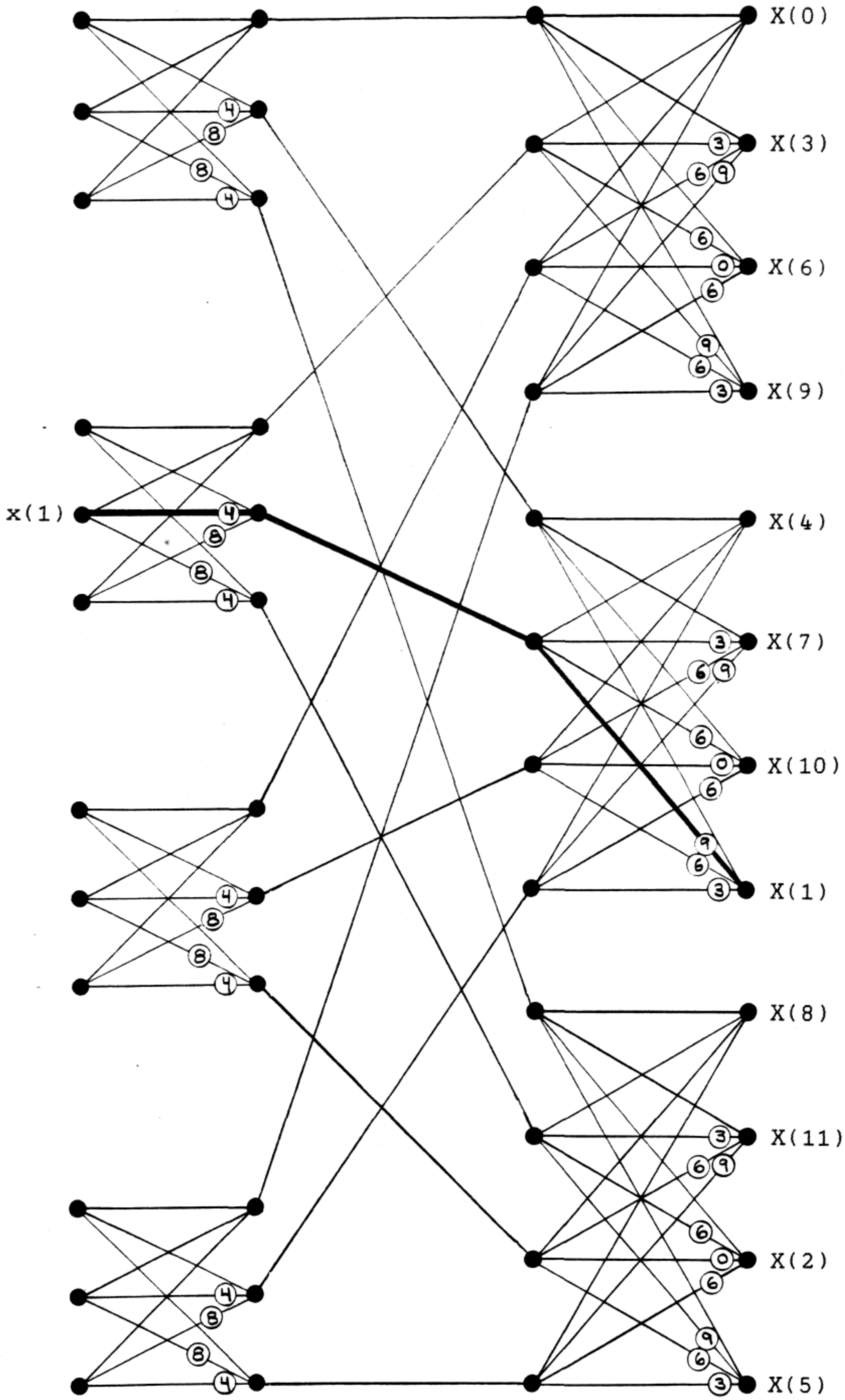

Fig. 2 Relating x(1) and X(1) determines output ordering by working left to right from x(1)

Having now established the interconnecting lines, and the position of X(0), we need to consider the remaining inputs and outputs. With regard to the x(n), we only know that each of the 12 x(n), one to a position, must be present on the left to serve in the formation of X(0). It will be useful here to first look for a possible path between x(1) at the input and X(1) at the output. That is, we want a path for n=1 and k=1, which of course means that the total "path twist" is nk=1. However, clearly, the only path twists directly available are 0, 3, 4, 6, 8, and 9. There is no 1. Accordingly, we look for a path which can produce a 1, if taken MOD-12. This can be provided by a total path twist of 13, available as a path twist of 4 from a 3-point DFT, plus a path twist of 9 from a 4-point DFT. Study of Fig. 2 indicates that there are four possible paths with a total twist of 13. We can choose any one of these at this point - it turns out that the other three choices provide more or less equivalent final results.

Choosing the position of x(1) as the middle input of the second 3-point DFT, we find X(1) as the lower output of the middle 4-point DFT (see Fig. 2). From this point on, the input and output orderings are fixed, and can be easily determined. For example, starting at x(1), we can follow a path across to some unknown X(k), keeping track of the total twists of the overall path. Since this total is nk, and since n=1 for x(1), the result must be due to k. For example, starting at x(1) on the left, and moving upward to the second output from the top on the right side, we encounter only a single path twist of 3, so this must be X(3). Similarly, starting at x(1) and going down to the bottom output on the right, we find a total path twist of 8 + 9 = 17. Since 17 MOD-12 is 5, this is X(5). Accordingly, the selection of x(1) determines the ordering at the output for all X(k).

In an exactly similar manner, the choice of X(1) on the right side (as constrained by the selected x(1) with a connecting overall path twist of 1) determines all the x(n). For example, starting at X(1) as shown in Fig. 3 and working back up phase-free paths only, we find that it is the upper left input point that must be x(0). Similarly, from the point X(1) back down to the lower left corner, we find a total path twist of 3 + 8 = 11, and the result identifies this lower corner as x(11).

The correctness of the final flow graph of Fig. 3 can be demonstrated, if necessary, by an exhaustive examination of all possible paths (144 of them in this case) to verify that they do in fact correspond to the 12-point DFT. Experience in this area will, however, generally lead to a belief in an "existence proof" so that once we find the last piece of our intuitive derivation to fall in place, we feel very confident that the whole thing works.

A few additional points can be made as we close. First, a study of the three flow graph that are alternative to Fig. 3 [different choices of x(1) and X(1)] shows no significant advantage in terms of input or output ordering. Secondly, while some FFT structures have additional path twists in the interconnecting lines, these were not considered in depth here, but appear to not offer any significant advantages either.

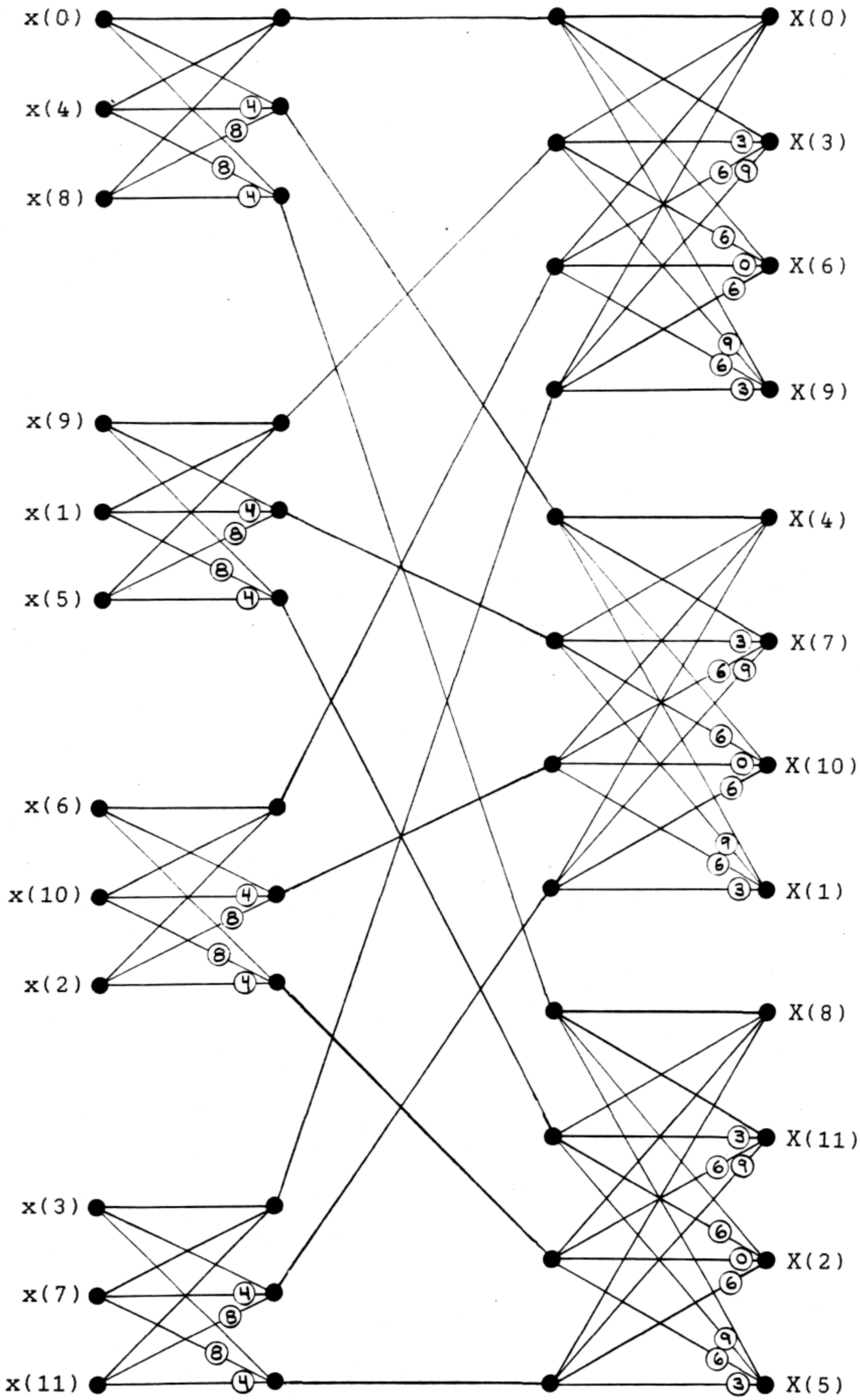

Fig. 3 Working back from X(1), right to left, determines input ordering, completing flow graph.

We also need to be sure to understand if this FFT really is advantageous, relative to the corresponding DFT. We can do this by counting all multiplies as being equal, regardless of whether or not they are fully complex, or simply multiplies by ±1 or ±j. In this case, the direct computation of the DFT would require 144 complex multiplies. A count of the multiplies in Fig. 3 (not counting the interconnects) gives 4 x 9 = 36 for the four 3-point DFT's and 3 x 16 = 48 for the three four-point DFT's. This is 84 multiplies total, and what seems to be a savings of 84/144.

Whether or not this is a savings of course depends on the relative "costs" of multiplies versus the cost of the "overhead" in Fig. 3. In Fig. 3, we need to do a good deal of such things as scrambling of input and output order, counting of blocks, interconnections, and so on. It is doubtful that the savings of multiplies would make up for the loss due to the FFT overhead for this 12-point case, although it is clear that FFT's will have significant advantages if we choose a high enough number of points.

It is known that Gauss (1777-1855) did use a 12-point FFT, and in his day, computation was of course done by hand. Accordingly, it is interesting to inquire about the efficiency of the FFT above if hand calculations were required. In such a case, it is clear that complex multiplications are fairly expensive, probably requiring several minutes each. In contrast, simply assigning the result to a scrambled location might only require a second. Further, the encounter of a path multiplier of 1, of -1, or even of +j or -j would have been significantly refreshing. Indeed, only 16 of the multiplies of Fig. 3 are non-trivial (the path twists of the 3-point DFT's). In hand computation, even the 12-point FFT would seem to be an astounding short cut.

This leads to a final point with regard to whether or not it would be worth our trouble to reduce the 4-point DFT's to a combination of 2-point DFT's. Since the 4-point DFT's involve only multiplies of ±1 or ±j, it would be a questionable advantage to reduce the 4-point DFT's further. For example, to multiply by j we only need to exchange the real and the imaginary parts of the complex number to be multiplied.